IBM Developing Applications Using IBM Informix ESQL/C

Duration: 4 Days       Course Code: IX150G

Overview:

This is the Classroom version of Instructor-Led Online course Developing Applications Using IBM Informix ESQL/C - Instructor Led Online (3X150) and Self-Paced Virtual course Developing Applications Using IBM Informix ESQL/C (SPVC) (2X150).

In this course, you will use IBM Informix ESQL/C tools to write applications that contain embedded SQL commands and queries. You will write applications to perform queries that return single and multiple rows; insert, update, and delete rows; create and use forms to display one or more rows of data; manage cursors, and handle various types of data, including large objects.

This course replaces US Course Developing Applications Using IBM Informix ESQL/C (L1112).

Target Audience:

This intermediate course is designed for application developers.

Objectives:

- Include SQL statements in a C program to add, retrieve, and alter data in an Informix database
- Set up and execute dynamically defined SQL statements
- Set up a cursor structure for manipulating a set of rows
- Write effective, efficient SQL-based applications in C
- Work with all SQL data types

Prerequisites:

You should have:

- IBM Informix Structured Query Language or equivalent knowledge
- ANSI C programming or C programming experience
Content:

Introduction to IBM Informix ESQL/C

- Identify the components of ESQL/C
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
IX150G

www.globalknowledge.com.sa training@globalknowledge.com.sa 00 966 92000 9278
The Demonstration Database

The tables in the stores_demo demonstration database

Install ESQ/C as part of Client SDK
Set environment variables and execute the ESQ/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQ/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQ/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQ/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the
Effectively use information contained in the structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were

- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

The Demonstration Database

- The tables in the stores_demo demonstration database

Install ESQ/C as part of Client SDK
Set environment variables and execute the ESQ/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQ/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQ/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQ/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the
Install ESL/C as part of Client SDK  
Set environment variables and execute the
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOF feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Deferred execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 14: Using Scroll Cursors Effectively
Exercise 14
Unit 15: Using Scroll Cursors
Exercise 15
Unit 16: Using Scroll Cursors
Exercise 16
Unit 17: Using an Update Cursor
Exercise 17
Unit 18: Using an Insert Cursor
Exercise 18
Unit 19: Time Data Types
Exercise 19
Unit 20: Working with the Database Server
Exercise 20
Unit 21: Dynamic SQL
Exercise 21
Unit 22: Dynamic SQL: Constructing
Exercise 22
Unit 23: Working with the Database Server
Exercise 23
BOOLEAN, and DECIMAL data types

- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldbbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  - Unit 17: Using an Update Cursor

### INSERT Statements

- Unit 23: Working with the Database Server
  - Exercise 19
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldbbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure of the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 14: Using Scroll Cursors
Exercise 14
Unit 15: Using Scroll Cursors Effectively
Exercise 15
Unit 16: Using an Update Cursor
Exercise 16
Unit 17: Using a Cursor
Exercise 17
Unit 18: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure of the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Using CONNECT TO INTERVAL macros

Describe the structure of an ESQL/C

SQL databases

Identify C variables for use in accessing SQL databases

Describe the structure of an ESQL/C

Using the CONNECT TO, DISCONNECT, and SET CONNECTION SQL statements

Install ESQL/C as part of Client SDK

Set environment variables and execute the ESQL/C preprocessor

The relationships between these tables

Installing a copy of this database

Identify C variables for use in accessing SQL databases

Describe the structure of an ESQL/C

Describe how the syntax works with pre-6.0 version syntax and functionality

Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the basic use of the make utility

Explain problems in converting between data types

Use functions to convert variables of different types

Explain the storage needs of character and string data

Interface with LVARCHAR data through library functions

Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types

Describe the structure for the DECIMAL data type

Use ESQL/C library functions to access data

Effectively use information contained in this structure

Simplify exception testing after every SQL statement

Describe how to obtain warning and error information

Determine whether SQL NULLs were fetched or character data was truncated

Ensure referential integrity using application logic

Use the appropriate cursor for a given application logic

Ensure referential integrity using application logic

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqlbreak

sqldetach

sqlbreak

Use functions to convert variables of selected rows

Use a scrolling cursor to browse the ESQL/C data

Describe how to obtain warning and error information

Determine whether SQL NULLs were fetched or character data was truncated

Ensure referential integrity using application logic

Use the appropriate cursor for a given task

Use a scrolling cursor to browse the selected rows

Change the size of FETCH and INSERT buffers

Automatically free a cursor

Use the OPTOFC feature to reduce network messaging

Solve the stale data problem by using the primary key to select the current row

Declare a cursor from a prepared statement

Defer execution of a PREPAREd statement

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqlbreak

sqlbreak

Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C

Exercise 1

Unit 2: The Demonstration Database

Exercise 2

Unit 3: Embedding SQL Statements

Exercise 3

Unit 4: Using CONNECT TO

Exercise 4

Unit 5: Compiling an ESQL/C Program

Exercise 5

Unit 6: ESQL/C Data Types

Unit 7: Character and String Data Types

Unit 8: Numeric Data Types

Unit 9: SQL Exception Testing

Exercise 10

Unit 12: Using GET DIAGNOSTICS

Exercise 11

Unit 13: Singleton Queries and Lookups

Exercise 12

Unit 14: Using Scroll Cursors

Exercise 13

Unit 15: Using Scroll Cursors Effectively

Unit 17: Using an Update Cursor

Exercise 15

Unit 18: Using an Insert Cursor

Unit 19: Time Data Types

Exercise 16

Unit 20: Dynamic SQL

Exercise 17

Unit 21: Dynamic SQL: Constructing INSERT Statements

Unit 22: Dynamic SQL: Constructing INSERT Statements

Unit 23: Working with the Database Server

Exercise 19

Using CONNECT TO

Use the CONNECT TO, DISCONNECT, and SET CONNECTION SQL statements

Install ESQL/C as part of Client SDK

Set environment variables and execute the ESQL/C preprocessor

The relationships between these tables

Installing a copy of this database

Identify C variables for use in accessing SQL databases

Describe the structure of an ESQL/C

Describe how the syntax works with pre-6.0 version syntax and functionality

Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the basic use of the make utility

Explain problems in converting between data types

Use functions to convert variables of different types

Explain the storage needs of character and string data

Interface with LVARCHAR data through library functions

Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types

Describe the structure for the DECIMAL data type

Use ESQL/C library functions to access data

Effectively use information contained in this structure

Simplify exception testing after every SQL statement

Describe how to obtain warning and error information

Determine whether SQL NULLs were fetched or character data was truncated

Ensure referential integrity using application logic

Use the appropriate cursor for a given task

Use a scrolling cursor to browse the selected rows

Change the size of FETCH and INSERT buffers

Automatically free a cursor

Use the OPTOFC feature to reduce network messaging

Solve the stale data problem by using the primary key to select the current row

Declare a cursor from a prepared statement

Defer execution of a PREPAREd statement

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqlbreak

sqlbreak

Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C

Exercise 1

Unit 2: The Demonstration Database

Exercise 2

Unit 3: Embedding SQL Statements

Exercise 3

Unit 4: Using CONNECT TO

Exercise 4

Unit 5: Compiling an ESQL/C Program

Exercise 5

Unit 6: ESQL/C Data Types

Unit 7: Character and String Data Types

Unit 8: Numeric Data Types

Unit 9: SQL Exception Testing

Exercise 10

Unit 12: Using GET DIAGNOSTICS

Exercise 11

Unit 13: Singleton Queries and Lookups

Exercise 12

Unit 14: Using Scroll Cursors

Exercise 13

Unit 15: Using Scroll Cursors Effectively

Unit 17: Using an Update Cursor

Exercise 15

Unit 18: Using an Insert Cursor

Unit 19: Time Data Types

Exercise 16
program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqlbreak
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  - Exercise 15
- Unit 17: Using an Update Cursor
  - Exercise 16
- Unit 18: Using an Insert Cursor
  - Exercise 17
- Unit 19: Time Data Types
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19

Using Scroll Cursors Effectively
- Use a scrolling cursor to select a primary key
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Exercise 19
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Ubuntu 23: Working with the Database Server
Exercise 6
Exercise 7
Unit 8: Numeric Data Types
Exercise 8
Unit 7: Character and String Data Types
Exercise 9
Exercise 10
Unit 10: The SQL Communications Area
Exercise 11
Unit 11: SQL Exception Testing
Exercise 12
Exercise 13
Unit 13: Singleton Queries and Lookups
Exercise 14
Exercise 19
Unit 15: Using Scroll Cursors
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFNC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldb
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 7: Character and String Data Types
  - Exercise 7
- Unit 8: Numeric Data Types
  - Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  - Exercise 15
- Unit 17: Using Scroll Cursors Effectively
  - Exercise 16
- Unit 18: Using an Insert Cursor
  - Exercise 17
- Unit 19: Time Data Types
  - Exercise 18
- Unit 21: Dynamic SQL
  - Exercise 19
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Unit 23: Working with the Database Server
Exercise 19

Compiling an ESQL/C Program

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLS were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects
Describe the structure for the DECIMAL macros
Exercise 5

Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 2: The Demonstration Database
Exercise 1
Unit 3: Embedding SQL Statements
Exercise 2
Unit 4: Using CONNECT TO
Exercise 3
Unit 5: Compiling an ESQL/C Program
Exercise 4
Unit 6: ESQL/C Data Types
Exercise 5
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Inserting Rows
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 19: Time Data Types
Exercise 18
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Exercise 14
Unit 15: Using Scroll Cursors
Exercise 13
Exercise 12
Unit 13: Singleton Queries and Lookups
Exercise 11
Exercise 10
Unit 11: SQL Exception Testing
Exercise 9
Unit 10: The SQL Communications Area
Exercise 8
Exercise 7
Unit 9: Inserting Rows
Exercise 6
Unit 8: Numeric Data Types
Exercise 5
Unit 7: Character and String Data Types
Exercise 4
Unit 6: ESQL/C Data Types
Exercise 3
Unit 5: Compiling an ESQL/C Program
Exercise 2
Unit 4: Using CONNECT TO
Exercise 1
Unit 3: Embedding SQL Statements

Index

Exercise 19

Compiling an ESQL/C Program

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLS were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects
Describe the structure for the DECIMAL macros

Indexes

Exercise 5

Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Day 2

Unit 2: The Demonstration Database
Exercise 1
Unit 3: Embedding SQL Statements
Exercise 2
Unit 4: Using CONNECT TO
Exercise 3
Unit 5: Compiling an ESQL/C Program
Exercise 4
Unit 6: ESQL/C Data Types
Exercise 5
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Inserting Rows
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 19: Time Data Types
Exercise 18
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Index
Use functions to convert variables of data types

Explain problems in converting between data types

Explain the basic use of the make utility

Describe when optional syntax is appropriate

Describe how the syntax works with pre-6.0 version syntax and functionality

Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the structure of an ESQL/C program

Describe how when optional syntax is appropriate

Describe how the syntax works with different types

Explain the storage needs of character and string data

Interface with VARCHAR data through library functions

Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types

Describe the structure for the DECIMAL data type

Use ESQL/C library functions to access data

Effectively use information contained in this structure

Simplify exception testing after every SQL statement

Describe how to obtain warning and error information

Determine whether SQL NULLs were fetched or character data was truncated

Ensure referential integrity using application logic

Use the appropriate cursor for a given task

Use a scrolling cursor to browse the selected rows

Change the size of FETCH and INSERT buffers

Automatically free a cursor

Use the OPTOFC feature to reduce network messaging

Solve the stale data problem by using the primary key to select the current row

Declare a cursor from a prepared statement

Defer execution of a PREPARED statement

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqldetach

sqldetach

Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C

Exercise 1

Unit 2: The Demonstration Database

Exercise 2

Unit 3: Embedding SQL Statements

Exercise 3

Unit 4: Using CONNECT TO

Exercise 4

Unit 5: Compiling an ESQL/C Program

Exercise 5

Unit 6: ESQL/C Data Types

Unit 7: Character and String Data Types

Exercise 6

Unit 8: Numeric Data Types

Exercise 7

Unit 9: Time Data Types

Exercise 8

Unit 10: The SQL Communications Area

Exercise 9

Unit 11: SQL Exception Testing

Exercise 10

Unit 12: Using GET DIAGNOSTICS

Exercise 11

Unit 13: Singleton Queries and Lookups

Exercise 12

Unit 14: Using an Update Cursor

Exercise 13

Unit 15: Using Scroll Cursors

Exercise 14

Unit 16: Using Scroll Cursors Effectively

Exercise 15

Unit 17: Using an Update Cursor

Exercise 16

Unit 18: Using an Insert Cursor

Unit 19: Time Data Types

Exercise 17

Unit 20: Dynamic SQL

Exercise 18

Unit 21: Dynamic SQL: Constructing INSERT Statements

Unit 22: Dynamic SQL: Constructing INSERT Statements

Unit 23: Working with the Database Server

Exercise 19

Install ESQL/C as part of Client SDK

Set environment variables and execute the ESQL/C preprocessor

The relationships between these tables

Installing a copy of this database

Identify C variables for use in accessing SQL databases

Describe the structure of an ESQL/C program

Describe when optional syntax is appropriate

Describe how the syntax works with pre-6.0 version syntax and functionality

Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the basic use of the make utility

Explain problems in converting between data types

Use functions to convert variables of data type

Use ESQL/C library functions to access data

Effectively use information contained in this structure

Simplify exception testing after every SQL statement

Describe how to obtain warning and error information

Determine whether SQL NULLs were fetched or character data was truncated

Ensure referential integrity using application logic

Use the appropriate cursor for a given task

Use a scrolling cursor to browse the selected rows

Change the size of FETCH and INSERT buffers

Automatically free a cursor

Use the OPTOFC feature to reduce network messaging

Solve the stale data problem by using the primary key to select the current row

Declare a cursor from a prepared statement

Defer execution of a PREPARED statement

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqldetach

sqldetach

Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C

Exercise 1

Unit 2: The Demonstration Database

Exercise 2

Unit 3: Embedding SQL Statements

Exercise 3

Unit 4: Using CONNECT TO

Exercise 4

Unit 5: Compiling an ESQL/C Program

Exercise 5

Unit 6: ESQL/C Data Types

Unit 7: Character and String Data Types

Exercise 6

Unit 8: Numeric Data Types

Exercise 7

Unit 9: Time Data Types

Exercise 8

Unit 10: The SQL Communications Area

Exercise 9

Unit 11: SQL Exception Testing

Exercise 10

Unit 12: Using GET DIAGNOSTICS

Exercise 11

Unit 13: Singleton Queries and Lookups

Exercise 12

Exercise 13

Unit 15: Using Scroll Cursors
different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
  - Unit 2: The Demonstration Database
  - Exercise 2
  - Unit 3: Embedding SQL Statements
  - Exercise 3
  - Unit 4: Using CONNECT TO
  - Exercise 4
  - Unit 5: Compiling an ESQL/C Program
  - Exercise 5
  - Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
  - Exercise 6
  - Unit 8: Numeric Data Types
  - Exercise 7
  - Exercise 8
  - Unit 10: The SQL Communications Area
  - Exercise 9
  - Unit 11: SQL Exception Testing
  - Exercise 10
  - Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
  - Unit 13: Singleton Queries and Lookups
  - Exercise 12
  - Exercise 13
  - Unit 15: Using Scroll Cursors
  - Exercise 14
  - Unit 16: Using Scroll Cursors Effectively
  - Unit 17: Using an Update Cursor
  - Exercise 15
  - Unit 18: Using an Insert Cursor
  - Unit 19: Time Data Types
  - Exercise 16
  - Exercise 17
  - Unit 21: Dynamic SQL
  - Exercise 18
  - Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Unit 23: Working with the Database Server
  - Exercise 19
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
ESQL/C Data Types

- Determine which C data type to use as a host variable
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Exercise 1
Exercise 3

Unit 3: Embedding SQL Statements

Exercise 2

Unit 2: The Demonstration Database

Exercise 1

Unit 1: Introduction to IBM Informix ESQL/C

---

ESQL/C

- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldbread
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  - Exercise 15
- Unit 17: Using an Update Cursor
  - Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
  - Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Exercise 19
- Unit 23: Working with the Database Server
  - Exercise 19

ESQL/C

- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldbread
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  - Exercise 15
- Unit 17: Using an Update Cursor
  - Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
  - Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Exercise 19
- Unit 23: Working with the Database Server
  - Exercise 19

Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
Use ESQL/C library functions to access data type BOOLEAN, and DECIMAL data types. Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types. Interface with LVARCHAR data through SERIAL8, BOOLEAN, and DECIMAL string data. Explain the storage needs of character and library functions. Use functions to convert variables of different types. Explain problems in converting between different types. Use a scrolling cursor to browse the selected rows. Effectively use information contained in this structure. Simplify exception testing after every SQL statement. Describe how to obtain warning and error information. Determine whether SQL NULLs were fetched or character data was truncated. Ensure referential integrity using application logic. Use the appropriate cursor for a given task. Use a scrolling cursor to browse the selected rows.
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqldbbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
  - Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
  - Unit 16: Using Scroll Cursors Effectively
  - Unit 17: Using an Update Cursor
  - Exercise 15
  - Unit 18: Using an Insert Cursor
  - Unit 19: Time Data Types
  - Exercise 16
- Exercise 19
Unit 21: Dynamic SQL
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing
INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Character and String Data Types

- Declare host variables for CHAR, CHAR *, VARCHAR, and LVARCHAR data types

Simple Large Objects

- Declare host variables for BYTE and TEXT data types

Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using CONNECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
- Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
- Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing
INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19
- Install ESQL/C as part of Client SDK
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 7: Character and String Data Types
  - Exercise 7
- Unit 8: Numeric Data Types
  - Exercise 8
- Unit 9: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Unit 14: Using a Scroll Cursor
  - Exercise 14
- Unit 15: Using Scroll Cursors Effectively
  - Exercise 16
- Unit 16: Using Scroll Cursors
  - Unit 17: Using an Update Cursor
  - Exercise 15
- Unit 18: Using an Insert Cursor
- Exercise 19
- Unit 19: Time Data Types
  - Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
  - Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Unit 23: Working with the Database Server
  - Exercise 19
- Install ESQL/C as part of Client SDK
  - Set environment variables and execute the ESQL/C preprocessor
  - The relationships between these tables
  - Installing a copy of this database
  - Identify C variables for use in accessing SQL databases
  - Describe the structure of an ESQL/C program
  - Describe when optional syntax is appropriate
  - Describe how the syntax works with pre-6.0 version syntax and functionality
- Change the size of FETCH and INSERT buffers
  - Automatically free a cursor
  - Use the OPTOFC feature to reduce network messaging
  - Solve the stale data problem by using the primary key to select the current row
  - Declare a cursor from a prepared statement
  - Defer execution of a PREPAREd statement
  - Use an INSERT cursor to insert rows into a database
  - Explain how data is converted as it is stored
  - Use pre-defined DATETIME and INTERVAL macros
  - Describe the locator structure
  - INSERT and SELECT simple large objects from a file or from memory
  - sqldetach
  - sqlbreak
  - Work with multiplexed connections
- Use functions to convert variables of different types
  - Explain the storage needs of character and string data
  - Interface with LVARCHAR data through library functions
  - Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
  - Describe the structure for the DECIMAL data type
  - Use ESQL/C library functions to access data
  - Effectively use information contained in this structure
  - Simplify exception testing after every SQL statement
  - Describe how to obtain warning and error information
  - Determine whether SQL NULLs were fetched or character data was truncated
  - Ensure referential integrity using application logic
  - Use the appropriate cursor for a given task
  - Use a scrolling cursor to browse the selected rows
  - Change the size of FETCH and INSERT buffers
  - Automatically free a cursor
  - Use the OPTOFC feature to reduce network messaging
  - Solve the stale data problem by using the primary key to select the current row
  - Declare a cursor from a prepared statement
  - Defer execution of a PREPAREd statement
  - Use an INSERT cursor to insert rows into a database
  - Explain how data is converted as it is stored
  - Use pre-defined DATETIME and INTERVAL macros
  - Describe the locator structure
  - INSERT and SELECT simple large objects from a file or from memory
  - sqldetach
  - sqlbreak
  - Work with multiplexed connections
Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the basic use of the make utility

Explain problems in converting between data types

Use functions to convert variables of different types

Explain the storage needs of character and string data

Interface with LVARCHAR data through library functions

Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types

Describe the structure for the DECIMAL data type

Use ESQL/C library functions to access data

Describe the locator structure

INSERT and SELECT simple large objects task

Defer execution of a PREPARED statement

Declare a cursor from a prepared statement

Solve the stale data problem by using the this structure

Use the OPTOFC feature to reduce network data

Automatic free a cursor

Use the OPTOFC feature to reduce network messaging

Solve the stale data problem by using the primary key to select the current row

Declare a cursor from a prepared statement

Defer execution of a PREPARED statement

Use an INSERT cursor to insert rows into a database

Explain how data is converted as it is stored

Use pre-defined DATETIME and INTERVAL macros

Describe the locator structure

INSERT and SELECT simple large objects from a file or from memory

sqldetach

sqlbreak

Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C

Exercise 1

Unit 2: The Demonstration Database

Exercise 2

Unit 3: Embedding SQL Statements

Exercise 3

Unit 4: Using CONNECT TO

Exercise 4

Unit 5: Compiling an ESQL/C Program

Exercise 5

Unit 6: ESQL/C Data Types

Exercise 6

Unit 7: Character and String Data Types

Unit 8: Numeric Data Types

Exercise 7

Exercise 8

Unit 9: SQL Communications Area

Exercise 9

Unit 10: SQL Exception Testing

Exercise 10

Unit 11: Using GET DIAGNOSTICS

Exercise 11

Unit 12: Using an Update Cursor

Exercise 12

Unit 13: Singleton Queries and Lookups

Exercise 13

Unit 14: Using Scroll Cursors

Exercise 14

Unit 15: Using Scroll Cursors Effectively

Exercise 15

Unit 16: Using an Insert Cursor

Exercise 16

Unit 17: Time Data Types

Exercise 17

Unit 18: Using an Update Cursor

Exercise 18

Unit 19: Dynamic SQL

Exercise 19

Install ESQL/C as part of Client SDK

Set environment variables and execute the ESQL/C preprocessor

The relationships between these tables

Installing a copy of this database

Identify C variables for use in accessing SQL databases

Describe the structure of an ESQL/C program

Describe when optional syntax is appropriate

Describe how the syntax works with pre-6.0 version syntax and functionality

Conditionally preprocess SQL statements

Compile a program by using IBM Informix ESQL/C

Explain the basic use of the make utility

Exercise 19

Explain problems in converting between data types

Use functions to convert variables of different types

Explain the storage needs of character and string data
Numeric Data Types

- Recognize the storage needs for numeric data

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
- Exercise 1
- Unit 2: The Demonstration Database
- Exercise 2
- Unit 3: Embedding SQL Statements
- Exercise 3
- Unit 4: Using SELECT TO
- Exercise 4
- Unit 5: Compiling an ESQL/C Program
- Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
- Exercise 6
- Unit 8: Numeric Data Types
- Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
- Exercise 9
- Unit 11: SQL Exception Testing
- Exercise 10
- Unit 12: Using GET DIAGNOSTICS
- Exercise 11
- Unit 13: Singleton Queries and Lookups
- Exercise 12
- Exercise 13
- Unit 14: Using Scroll Cursors
- Exercise 14
- Unit 15: Using Scroll Cursors Effectively
- Exercise 15
- Unit 16: Using Scroll Cursors
- Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
- Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
- Exercise 19
- Dynamic SQL
- Use dynamic SQL and the associated data structures and commands
- Dynamic SQL: Constructing INSERT Statements
- Use dynamic SQL to construct insert statements at runtime
- Working with the Database Server
- Explain how to control the database server process with these functions:
  - sqlexit
  - sqldetach
  - sqlbreak
  - Work with multiplexed connections
  - Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
  - Describe the structure for the DECIMAL data type
  - Use ESQL/C library functions to access data
  - Effectively use information contained in this structure
  - Simplify exception testing after every SQL statement
  - Describe how to obtain warning and error information
  - Determine whether SQL NULLs were fetched or character data was truncated
  - Ensure referential integrity using application logic
  - Use the appropriate cursor for a given task
  - Use a scrolling cursor to browse the selected rows
  - Change the size of FETCH and INSERT buffers
  - Automatically free a cursor
  - Use the OPTOFC feature to reduce network messaging
  - Solve the stale data problem by using the primary key to select the current row
  - Declare a cursor from a prepared statement
  - Defer execution of a PREPAREd statement
  - Use an INSERT cursor to insert rows into a database
  - Explain how data is converted as it is stored
  - Use pre-defined DATETIME and INTERVAL macros
  - Describe the locator structure
  - INSERT and SELECT simple large objects from a file or from memory
Exercise 19
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Time Data Types
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Time Data Types
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Time Data Types
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
Unit 23: Working with the Database Server
Exercise 19

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C

Describe how the structure works with pre-6.0 version syntax and functionality
Describe when optional syntax is appropriate
Describe when optional syntax is appropriate
Describe the structure of an ESQL/C program

Objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Unit 14: Using Scroll Cursors
Exercise 13
Unit 15: Using Scroll Cursors Effectively
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Use functions to convert variables of different types
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Unit 15: Using Scroll Cursors Effectively
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19
<table>
<thead>
<tr>
<th>Exercise 6</th>
<th>Exercise 5</th>
<th>Exercise 4</th>
<th>Exercise 3</th>
<th>Exercise 2</th>
<th>Exercise 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 3: Embedding SQL Statements</td>
<td>Unit 4: Using CONNECT TO</td>
<td>Unit 5: Compiling an ESQ/C Program</td>
<td>Unit 6: ESQ/C Data Types</td>
<td>Unit 7: Character and String Data Types</td>
<td>Unit 1: Introduction to IBM Informix ESQ/C</td>
</tr>
<tr>
<td>Exercise 10</td>
<td>Exercise 9</td>
<td>Exercise 8</td>
<td>Exercise 7</td>
<td>Exercise 6</td>
<td>Exercise 1</td>
</tr>
<tr>
<td>Unit 10: The SQL Communications Area</td>
<td>Unit 11: SQL Exception Testing</td>
<td>Unit 12: Using GET DIAGNOSTICS</td>
<td>Unit 13: Singleton Queries and Lookups</td>
<td>Unit 14: Using Scroll Cursors</td>
<td>Unit 15: Using Scroll Cursors</td>
</tr>
<tr>
<td>Exercise 16</td>
<td>Exercise 15</td>
<td>Exercise 14</td>
<td>Exercise 13</td>
<td>Exercise 12</td>
<td>Exercise 11</td>
</tr>
<tr>
<td>Unit 18: Using an Insert Cursor</td>
<td>Unit 19: Time Data Types</td>
<td>Exercise 16</td>
<td>Exercise 17</td>
<td>Unit 21: Dynamic SQL</td>
<td>Unit 22: Dynamic SQL: Constructing INSERT Statements</td>
</tr>
<tr>
<td>Exercise 19</td>
<td>Exercise 18</td>
<td>Unit 23: Working with the Database Server</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exercise 3</th>
<th>Exercise 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 4: Using CONNECT TO</td>
<td>Unit 5: Compiling an ESQ/C Program</td>
</tr>
<tr>
<td>Exercise 4</td>
<td>Exercise 3</td>
</tr>
<tr>
<td>Unit 3: Embedding SQL Statements</td>
<td>Unit 2: The Demonstration Database</td>
</tr>
<tr>
<td>Exercise 2</td>
<td>Exercise 1</td>
</tr>
<tr>
<td>Unit 1: Introduction to IBM Informix ESQ/C</td>
<td></td>
</tr>
</tbody>
</table>
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqlbreak
sqldetach
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Exercise 6
Unit 7: Character and String Data Types
Exercise 7
Unit 8: Numeric Data Types
Exercise 8
Unit 9: Time Data Types
Exercise 9
Unit 10: The SQL Communications Area
Exercise 10
Unit 11: SQL Exception Testing
Exercise 11
Unit 12: Using GET DIAGNOSTICS
Exercise 12
Unit 13: Singleton Queries and Lookups
Exercise 13
Unit 14: An Overview of Cursors
Exercise 14
Unit 15: Using Scroll Cursors
Exercise 15
Unit 16: Using Scroll Cursors Effectively
Exercise 16
Unit 17: Using an Update Cursor
Exercise 17
Unit 18: Using an Insert Cursor
Exercise 18
Unit 19: Time Data Types
Exercise 19
Unit 20: Dynamic SQL
Exercise 20
Unit 21: Dynamic SQL: Constructing INSERT Statements
Exercise 21
Unit 22: Dynamic SQL: Constructing INSERT Statements
Exercise 22
Unit 23: Working with the Database Server
Exercise 23
Day 1
Welcome
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
sqlbreak
sqldetach
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Exercise 6
Unit 7: Character and String Data Types
Exercise 7
Unit 8: Numeric Data Types
Exercise 8
Unit 9: Time Data Types
Exercise 9
Unit 10: The SQL Communications Area
Exercise 10
Unit 11: SQL Exception Testing
Exercise 11
Unit 12: Using GET DIAGNOSTICS
Exercise 12
Unit 13: Singleton Queries and Lookups
Exercise 13
Unit 14: An Overview of Cursors
Exercise 14
Unit 15: Using Scroll Cursors
Exercise 15
Unit 16: Using Scroll Cursors Effectively
Exercise 16
Unit 17: Using an Update Cursor
Exercise 17
Unit 18: Using an Insert Cursor
Exercise 18
Unit 19: Time Data Types
Exercise 19
Unit 20: Dynamic SQL
Exercise 20
Unit 21: Dynamic SQL: Constructing INSERT Statements
Exercise 21
Unit 22: Dynamic SQL: Constructing INSERT Statements
Exercise 22
Unit 23: Working with the Database Server
Exercise 23
Day 3
Unit 14: An Overview of Cursors
Exercise 14
Unit 15: Using Scroll Cursors
Exercise 15
Unit 16: Using Scroll Cursors Effectively
Exercise 16
Unit 17: Using an Update Cursor
Exercise 17
Unit 18: Using an Insert Cursor
Exercise 18
Unit 19: Time Data Types
Exercise 19
Unit 20: Dynamic SQL
Exercise 20
Unit 21: Dynamic SQL: Constructing INSERT Statements
Exercise 21
Unit 22: Dynamic SQL: Constructing INSERT Statements
Exercise 22
Unit 23: Working with the Database Server
Exercise 23
The SQL Communications Area

Inserting Rows

Code an appropriate embedded SQL statement that inserts a row into a database

The SQL Communications Area

Explain the need for and the structure of the SQL Communication Area

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through string data
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through string data
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through string data
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through string data
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqlbreak
sqldetach
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 9: Date and Time Data Types
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Unit 14: Using Scroll Cursors
Exercise 13
Unit 15: Using Scroll Cursors Effectively
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Exercise 15
Unit 17: Using an Update Cursor
Exercise 16
Unit 18: Using an Insert Cursor
Exercise 17
Unit 19: Time Data Types
Exercise 18
Unit 20: Dynamic SQL
Unit 21: Dynamic SQL: Constructing INSERT Statements
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

SQL Exception Testing
- Describe the kinds of exceptions that might occur during execution of an SQL statement.

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  Exercise 1
- Unit 2: The Demonstration Database
  Exercise 2
- Unit 3: Embedding SQL Statements
  Exercise 3
- Unit 4: Using CONNECT TO
  Exercise 4
- Unit 5: Compiling an ESQL/C Program
  Exercise 5
- Unit 6: ESQL/C Data Types
  Exercise 6
- Unit 7: Character and String Data Types
  Exercise 7
- Unit 8: Numeric Data Types
  Exercise 8
- Unit 10: The SQL Communications Area
  Exercise 9
- Unit 11: SQL Exception Testing
  Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  Exercise 11
- Unit 13: Singleton Queries and Lookups
  Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  Exercise 15
- Unit 17: Using an Update Cursor
  Exercise 16
- Unit 18: Using an Insert Cursor
  Exercise 17
- Unit 19: Time Data Types
  Exercise 18
- Unit 21: Dynamic SQL
  Exercise 19
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  Unit 23: Working with the Database Server
  Exercise 19
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Use interface functions to access data
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure

Describe the preferred error detection methods

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Use interface functions to access data
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
    - Exercise 6
    - Exercise 7
    - Exercise 8
    - Unit 10: The SQL Communications Area
      - Exercise 9
    - Unit 11: SQL Exception Testing
      - Exercise 10
    - Unit 12: Using GET DIAGNOSTICS
      - Exercise 11
    - Unit 13: Singleton Queries and Lookups
      - Exercise 12
      - Exercise 13
      - Unit 15: Using Scroll Cursors
        - Exercise 14
      - Unit 16: Using Scroll Cursors Effectively
        - Unit 17: Using an Update Cursor
          - Exercise 15
        - Unit 18: Using an Insert Cursor
          - Unit 19: Time Data Types
            - Exercise 16
            - Exercise 17
            - Unit 21: Dynamic SQL
              - Exercise 18
            - Unit 22: Dynamic SQL: Constructing INSERT Statements
              - Unit 23: Working with the Database Server
                - Exercise 19
  - Install ESQL/C as part of Client SDK
  - Set environment variables and execute the ESQL/C preprocessor
    - The relationships between these tables
    - Installing a copy of this database
    - Identify C variables for use in accessing SQL databases
    - Describe the structure of an ESQL/C program
    - Describe when optional syntax is appropriate
    - Describe how the syntax works with pre-6.0 version syntax and functionality
    - Conditionally preprocess SQL statements
    - Compile a program by using IBM Informix ESQL/C
  - Explain how data is converted as it is stored
  - Use pre-defined DATETIME and INTERVAL macros
  - Describe the locator structure
  - INSERT and SELECT simple large objects from a file or from memory
  - sqldetach
  - sqlbreak
  - Work with multiplexed connections
  - Unit 1: Introduction to IBM Informix ESQL/C
    - Exercise 1
  - Unit 2: The Demonstration Database
    - Exercise 2
  - Unit 3: Embedding SQL Statements
    - Exercise 3
  - Unit 4: Using CONNECT TO
    - Exercise 4
  - Unit 5: Compiling an ESQL/C Program
    - Exercise 5
  - Unit 6: ESQL/C Data Types
    - Unit 7: Character and String Data Types
      - Exercise 6
      - Exercise 7
      - Exercise 8
      - Unit 10: The SQL Communications Area
        - Exercise 9
      - Unit 11: SQL Exception Testing
        - Exercise 10
      - Unit 12: Using GET DIAGNOSTICS
        - Exercise 11
      - Unit 13: Singleton Queries and Lookups
        - Exercise 12
        - Exercise 13
        - Unit 15: Using Scroll Cursors
          - Exercise 14
        - Unit 16: Using Scroll Cursors Effectively
          - Unit 17: Using an Update Cursor
            - Exercise 15
        - Unit 18: Using an Insert Cursor
          - Unit 19: Time Data Types
            - Exercise 16
            - Exercise 17
            - Unit 21: Dynamic SQL
              - Exercise 18
            - Unit 22: Dynamic SQL: Constructing INSERT Statements
              - Unit 23: Working with the Database Server
                - Exercise 19
  - Install ESQL/C as part of Client SDK
  - Set environment variables and execute the ESQL/C preprocessor
    - The relationships between these tables
    - Installing a copy of this database
    - Identify C variables for use in accessing SQL databases
    - Describe the structure of an ESQL/C program
    - Describe when optional syntax is appropriate
    - Describe how the syntax works with
Singleton Queries and Lookups

Use the SELECT statement to retrieve a row from your database

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Describe the storage needs of character and string data
Interface with VARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqlbreak
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Describe the storage needs of character and string data
Interface with VARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqlbreak
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Exercise 19
Unit 23: Working with the Database Server
INSERT Statements
Unit 22: Dynamic SQL: Constructing data types
Exercise 18
Unit 21: Dynamic SQL
Exercise 17
Exercise 16
Exercise 15
Unit 18: Using an Insert Cursor
Exercise 14
Unit 17: Using an Update Cursor
Exercise 13
Exercise 12
Unit 13: Singleton Queries and Lookups
Exercise 11
Unit 12: Using GET DIAGNOSTICS
Exercise 10
Exercise 9
Unit 11: SQL Exception Testing
Unit 10: The SQL Communications Area
Exercise 8
Unit 8: Numeric Data Types
Exercise 7
Unit 7: Character and String Data Types
Exercise 6
Unit 6: ESQL/C Data Types
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 4: Using CONNECT TO
Exercise 4
Unit 3: Embedding SQL Statements
Exercise 3
Unit 2: The Demonstration Database
Exercise 2
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1

Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqlbreak
sqldetach
Work with multiplexed connections
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Exercise 15
Unit 17: Using an Update Cursor
Exercise 16
Unit 18: Using an Insert Cursor
Exercise 17
Unit 19: Time Data Types
Exercise 18
Unit 21: Dynamic SQL
Exercise 19
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
  - Unit 2: The Demonstration Database
  - Exercise 2
  - Unit 3: Embedding SQL Statements
  - Exercise 3
  - Unit 4: Using CONNECT TO
  - Exercise 4
  - Unit 5: Compiling an ESQL/C Program
  - Exercise 5
  - Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
  - Exercise 6
  - Unit 8: Numeric Data Types
  - Exercise 7
  - Exercise 8
  - Unit 10: The SQL Communications Area
  - Exercise 9
  - Unit 11: SQL Exception Testing
  - Exercise 10
  - Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
  - Unit 13: Singleton Queries and Lookups
  - Exercise 12
  - Exercise 13
  - Unit 15: Using Scroll Cursors
  - Exercise 14
  - Unit 16: Using Scroll Cursors Effectively
  - Unit 17: Using an Update Cursor
  - Exercise 15
  - Unit 18: Using an Insert Cursor
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Use functions to convert variables of different types
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Defer execution of a PREPAREd statement
primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix
  - ESQL/C
  - Exercise 1
  - Unit 2: The Demonstration Database
  - Exercise 2
  - Unit 3: Embedding SQL Statements
  - Exercise 3
  - Unit 4: Using CONNECT TO
  - Exercise 4
  - Unit 5: Compiling an ESQL/C Program
  - Exercise 5
  - Unit 6: ESQL/C Data Types
  - Exercise 6
  - Unit 7: Character and String Data Types
  - Exercise 7
  - Unit 8: Numeric Data Types
  - Exercise 8
  - Unit 10: The SQL Communications Area
  - Exercise 9
  - Unit 11: SQL Exception Testing
  - Exercise 10
  - Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
  - Unit 13: Singleton Queries and Lookups
  - Exercise 12
  - Unit 14: Singleton Queries and Lookups Effectively
  - Exercise 13
  - Unit 15: Using Scroll Cursors
  - Exercise 14
  - Unit 16: Using Scroll Cursors Effectively
  - Exercise 15
  - Unit 17: Using an Update Cursor
  - Exercise 16
  - Unit 18: Using an Insert Cursor
  - Exercise 17
  - Unit 19: Time Data Types
  - Exercise 18
  - Unit 20: Dynamic SQL
  - Exercise 19
  - Unit 21: Dynamic SQL: Constructing INSERT Statements
  - Exercise 20
  - Unit 22: Dynamic SQL: Constructing INSERT Statements
  - Exercise 21
  - Unit 23: Working with the Database Server
  - Exercise 19
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Install ESQL/C as part of Client SDK
- Set environment variables and execute the ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
ESQL/C
  Exercise 1
  Unit 2: The Demonstration Database
  Exercise 2
  Unit 3: Embedding SQL Statements
  Exercise 3
  Unit 4: Using CONNECT TO
  Exercise 4
  Unit 5: Compiling an ESQL/C Program
  Exercise 5
  Unit 6: ESQL/C Data Types
  Exercise 6
  Unit 7: Character and String Data Types
  Exercise 7
  Unit 8: Numeric Data Types
  Exercise 8
  Unit 10: The SQL Communications Area
  Exercise 9
  Unit 11: SQL Exception Testing
  Exercise 10
  Unit 12: Using GET DIAGNOSTICS
  Exercise 11
  Unit 13: Singleton Queries and Lookups
  Exercise 12
  Exercise 13
  Unit 15: Using Scroll Cursors
  Exercise 14
  Unit 16: Using Scroll Cursors Effectively
  Unit 17: Using an Update Cursor
  Exercise 15
  Unit 18: Using an Insert Cursor
  Unit 19: Time Data Types
  Exercise 16
  Exercise 17
  Unit 21: Dynamic SQL
  Exercise 18
  Unit 22: Dynamic SQL: Constructing INSERT Statements
  Unit 23: Working with the Database Server
  Exercise 19

Exercise 4
  Unit 5: Compiling an ESQL/C Program
  Exercise 5
  Unit 6: ESQL/C Data Types
  Exercise 6
  Unit 7: Character and String Data Types
  Exercise 7
  Unit 8: Numeric Data Types
  Exercise 8
  Unit 10: The SQL Communications Area
  Exercise 9
  Unit 11: SQL Exception Testing
  Exercise 10
  Unit 12: Using GET DIAGNOSTICS
  Exercise 11
  Unit 13: Singleton Queries and Lookups
  Exercise 12
  Exercise 13
  Unit 15: Using Scroll Cursors
  Exercise 14
  Unit 16: Using Scroll Cursors Effectively
  Unit 17: Using an Update Cursor
  Exercise 15
  Unit 18: Using an Insert Cursor
  Unit 19: Time Data Types
  Exercise 16
  Exercise 17
  Unit 21: Dynamic SQL
  Exercise 18
  Unit 22: Dynamic SQL: Constructing INSERT Statements
  Unit 23: Working with the Database Server
  Exercise 19

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocess or
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFCE feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Exercise 6
- Unit 7: Character and String Data Types
  - Exercise 7
- Unit 8: Numeric Data Types
  - Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
 Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
- Unit 7: Character and String Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
  - Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
- Unit 11: SQL Exception Testing
  - Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  - Exercise 11
- Unit 13: Singleton Queries and Lookups
  - Exercise 12
  - Exercise 13
- Unit 15: Using Scroll Cursors
  - Exercise 14
- Unit 16: Using Scroll Cursors Effectively
- Unit 17: Using an Update Cursor
  - Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
  - Exercise 16
  - Exercise 17
- Unit 21: Dynamic SQL
  - Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
- Unit 23: Working with the Database Server
  - Exercise 19

Day 4

- Unit 20: Simple Large Objects

- Install ESQL/C as part of Client SDK
- Set environment variables and execute the
ESQL/C preprocessor
- The relationships between these tables
- Installing a copy of this database
- Identify C variables for use in accessing SQL databases
- Describe the structure of an ESQL/C program
- Describe when optional syntax is appropriate
- Describe how the syntax works with pre-6.0 version syntax and functionality
- Conditionally preprocess SQL statements
- Compile a program by using IBM Informix ESQL/C
- Explain the basic use of the make utility
- Explain problems in converting between data types
- Use functions to convert variables of different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix
ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Exercise 19

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPAREd statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  Exercise 1
- Unit 2: The Demonstration Database
  Exercise 2
- Unit 3: Embedding SQL Statements
  Exercise 3
- Unit 4: Using CONNECT TO
  Exercise 4
- Unit 5: Compiling an ESQL/C Program
  Exercise 5
- Unit 6: ESQL/C Data Types
  Exercise 6
- Unit 7: Character and String Data Types
  Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  Exercise 9
- Unit 11: SQL Exception Testing
  Exercise 10
- Unit 12: Using GET DIAGNOSTICS
  Exercise 11
- Unit 13: Singleton Queries and Lookups
  Exercise 12
- Exercise 13
- Unit 15: Using Scroll Cursors
  Exercise 14
- Unit 16: Using Scroll Cursors Effectively
  Unit 17: Using an Update Cursor
  Exercise 15
- Unit 18: Using an Insert Cursor
- Unit 19: Time Data Types
  Exercise 16
- Exercise 17
- Unit 21: Dynamic SQL
  Exercise 18
- Unit 22: Dynamic SQL: Constructing INSERT Statements
  Unit 23: Working with the Database Server
  Exercise 19
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPAREd statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
Work with multiplexed connections

Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1

Unit 2: The Demonstration Database
Exercise 2

Unit 3: Embedding SQL Statements
Exercise 3

Unit 4: Using CONNECT TO
Exercise 4

Unit 5: Compiling an ESQL/C Program
Exercise 5

Unit 6: ESQL/C Data Types
Exercise 6

Unit 7: Character and String Data Types
Exercise 7

Unit 8: Numeric Data Types
Exercise 8

Unit 9: The SQL Communications Area
Exercise 9

Unit 10: SQL Exception Testing
Exercise 10

Unit 11: Using GET DIAGNOSTICS
Exercise 11

Unit 12: Singleton Queries and Lookups
Exercise 12

Unit 13: Using Scroll Cursors
Exercise 13

Unit 14: Using Scroll Cursors Effectively
Exercise 14

Unit 15: Using an Update Cursor
Exercise 15

Unit 16: Using an Insert Cursor
Exercise 16

Unit 17: Time Data Types
Exercise 17

Unit 18: Dynamic SQL
Exercise 18

Unit 19: Dynamic SQL: Constructing INSERT Statements
Exercise 19

Unit 20: Working with the Database Server

Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of
different types
- Explain the storage needs of character and string data
- Interface with LVARCHAR data through library functions
- Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
- Describe the structure for the DECIMAL data type
- Use ESQL/C library functions to access data
- Effectively use information contained in this structure
- Simplify exception testing after every SQL statement
- Describe how to obtain warning and error information
- Determine whether SQL NULLs were fetched or character data was truncated
- Ensure referential integrity using application logic
- Use the appropriate cursor for a given task
- Use a scrolling cursor to browse the selected rows
- Change the size of FETCH and INSERT buffers
- Automatically free a cursor
- Use the OPTOFC feature to reduce network messaging
- Solve the stale data problem by using the primary key to select the current row
- Declare a cursor from a prepared statement
- Defer execution of a PREPARED statement
- Use an INSERT cursor to insert rows into a database
- Explain how data is converted as it is stored
- Use pre-defined DATETIME and INTERVAL macros
- Describe the locator structure
- INSERT and SELECT simple large objects from a file or from memory
- sqldetach
- sqlbreak
- Work with multiplexed connections
- Unit 1: Introduction to IBM Informix ESQL/C
  - Exercise 1
- Unit 2: The Demonstration Database
  - Exercise 2
- Unit 3: Embedding SQL Statements
  - Exercise 3
- Unit 4: Using CONNECT TO
  - Exercise 4
- Unit 5: Compiling an ESQL/C Program
  - Exercise 5
- Unit 6: ESQL/C Data Types
  - Unit 7: Character and String Data Types
  - Exercise 6
- Unit 8: Numeric Data Types
  - Exercise 7
- Exercise 8
- Unit 10: The SQL Communications Area
  - Exercise 9
Install ESQL/C as part of Client SDK
Set environment variables and execute the ESQL/C preprocessor
The relationships between these tables
Installing a copy of this database
Identify C variables for use in accessing SQL databases
Describe the structure of an ESQL/C program
Describe when optional syntax is appropriate
Describe how the syntax works with pre-6.0 version syntax and functionality
Conditionally preprocess SQL statements
Compile a program by using IBM Informix ESQL/C
Explain the basic use of the make utility
Explain problems in converting between data types
Use functions to convert variables of different types
Explain the storage needs of character and string data
Interface with LVARCHAR data through library functions
Declare host variables for INT8, SERIAL8, BOOLEAN, and DECIMAL data types
Describe the structure for the DECIMAL data type
Use ESQL/C library functions to access data
Effectively use information contained in this structure
Simplify exception testing after every SQL statement
Describe how to obtain warning and error information
Determine whether SQL NULLs were fetched or character data was truncated
Ensure referential integrity using application logic
Use the appropriate cursor for a given task
Use a scrolling cursor to browse the selected rows
Change the size of FETCH and INSERT buffers
Automatically free a cursor
Use the OPTOFC feature to reduce network messaging
Solve the stale data problem by using the primary key to select the current row
Declare a cursor from a prepared statement
Defer execution of a PREPARED statement
Use an INSERT cursor to insert rows into a database
Explain how data is converted as it is stored
Use pre-defined DATETIME and INTERVAL macros
Describe the locator structure
INSERT and SELECT simple large objects from a file or from memory
sqldetach
sqlbreak
Work with multiplexed connections
Unit 1: Introduction to IBM Informix ESQL/C
Exercise 1
Unit 2: The Demonstration Database
Exercise 2
Unit 3: Embedding SQL Statements
Exercise 3
Unit 4: Using CONNECT TO
Exercise 4
Unit 5: Compiling an ESQL/C Program
Exercise 5
Unit 6: ESQL/C Data Types
Unit 7: Character and String Data Types
Exercise 6
Unit 8: Numeric Data Types
Exercise 7
Exercise 8
Unit 10: The SQL Communications Area
Exercise 9
Unit 11: SQL Exception Testing
Exercise 10
Unit 12: Using GET DIAGNOSTICS
Exercise 11
Unit 13: Singleton Queries and Lookups
Exercise 12
Exercise 13
Unit 15: Using Scroll Cursors
Exercise 14
Unit 16: Using Scroll Cursors Effectively
Unit 17: Using an Update Cursor
Exercise 15
Unit 18: Using an Insert Cursor
Unit 19: Time Data Types
Exercise 16
Exercise 17
Unit 21: Dynamic SQL
Exercise 18
Unit 22: Dynamic SQL: Constructing INSERT Statements
Unit 23: Working with the Database Server
Further Information:
For More information, or to book your course, please call us on 00 966 92000 9278
training@globalknowledge.com.sa
www.globalknowledge.com.sa

Global Knowledge - KSA, 393 Al-Uroubah Road, Al Worood, Riyadh 3140, Saudi Arabia