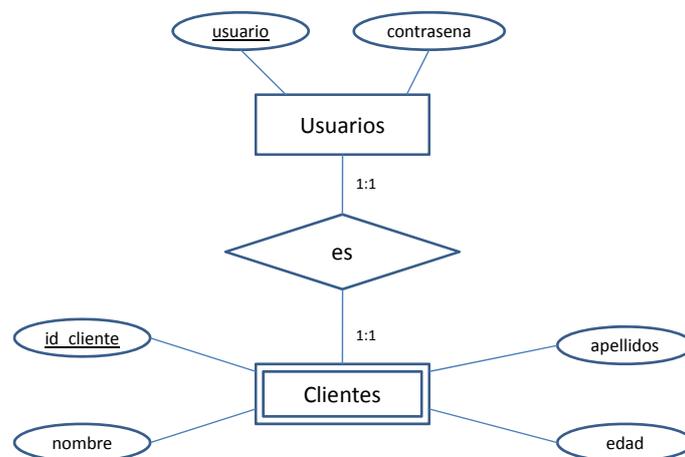


Seminario 1. Conexión a Oracle

Laboratorio de Programación
Curso 2007/2008
Fran J. Ruiz
Emilio Sobreviola

Diseño E/R



Creación de Tablas en SQL

`cliente.sql`

```
CREATE TABLE Usuarios (
  usuario      varchar(8) NOT NULL UNIQUE,
  contraseña   varchar(15) NOT NULL UNIQUE,
  CONSTRAINT Usuarios_PK PRIMARY KEY (usuario)
);

CREATE TABLE Clientes (
  id_cliente   varchar(8) NOT NULL UNIQUE,
  nombre       varchar(20) NOT NULL,
  apellidos    varchar(20) NOT NULL,
  edad         number(3),
  CONSTRAINT Clientes_FK FOREIGN KEY (id_cliente)
  REFERENCES Usuarios (usuario)
  ON DELETE CASCADE
}
```

Referencias para Pro*C

- Manuales de Referencia Oracle
(<http://www.oracle.com/pls/db102/homepage>)
 - Pro*C/C++ Precompiler Getting Started for Windows
 - Pro*C/C++ Precompiler Programmer's Guide
 - SQL*Plus Reference
 - PL/SQL User's Guide and Reference
- Ejemplos para Pro*C
 - Desde Hendrix:
`/usr/local/oracle/app/oracle/product/8.1.6/precomp/demo/proc`

Preámbulo

Insertar ficheros externos de cabecera (.h) en el código fuente:

```
/** Librería de Entrada Salida Estándar */
#include <stdio.h>

/** Manipulación de Strings */
#include <string.h>

/* Funciones estándar: atoi, rand, malloc, abs, size_t (tipo de datos) */
#include <stdlib.h>

/* Estructura de datos para SQL dinámica */
#include <sqlda.h>

/* Funciones SQLLIB generadas por Pro*C, para comunicarse con la interfaz
de comunicación de sentencias SQL a Oracle */
#include <sqlcpr.h>

/* Incluye SQL Communications Area, que permite ejecutar sentencias del
tipo EXEC SQL */
#include <sqlca.h>
```

Constantes, Variables y Estructuras

```
/** Declaración de Constantes (como directivas de precompilador) */
#define UNAME_LEN      20 // La longitud máxima de la cadena usuario
#define PWD_LEN        40 // La longitud máxima de la cadena passwd

/** Declaración de Variables */
VARCHAR username[UNAME_LEN];
VARCHAR password[PWD_LEN];
VARCHAR host[PWD_LEN];

/* Declaración de Estructuras. Por ejemplo, para el resultado de una SQL
SELECT */
struct{
    VARCHAR      nombre[UNAME_LEN];
    VARCHAR      apellidos[UNAME_LEN];
    short        edad;
} tuplaCliente;

/* Variables del programa C */
int             num_consultas;
```

Conexión con la Base de Datos

```
void conectarOracle(char *usuario, char *contrasena)
{
/* Conexión a Oracle
 * Copiar el nombre de usuario a la variable 'username' y establecer la
longitud */
    strncpy((char *) username.arr, usuario, UNAME_LEN);
    username.len = strlen((char *) username.arr);

/* Copiar la contraseña a la variable 'password' */
    strncpy((char *) password.arr, contrasena, PWD_LEN);
    password.len = strlen((char *) password.arr);

/* Copiar el host a la variable 'host' */
    strncpy((char *) host.arr, "vicious.den.cps.unizar.es", PWD_LEN);
    host.len = strlen((char *) host.arr);

/* Registrar la función sql_error() como capturador de eventos. */
    EXEC SQL WHENEVER SQLERROR DO sql_error("Error en Oracle:\n");

/* Conectar a Oracle. El programa llama a sql_error()
 * si sucede algún error en la conexión a la base de datos por defecto */
    EXEC SQL CONNECT :username IDENTIFIED BY :password USING :host;
}
}
```

Desconexión de la Base de Datos

```
void desconectarOracle()
{
/* Ejecutar las SQL pendientes (que estén en el buffer */
    EXEC SQL COMMIT WORK RELEASE;

    exit(0);
}
}
```

Consultas de Selección

```
void obtenerCliente(char *cliente)
{
    /* Crear la variable que contenga el identificador del cliente para la comunicación con la
    * base de datos */
    EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR          uid[UNAME_LEN];
    EXEC SQL END DECLARE SECTION;

    /* Copiar el identificador del cliente a la variable 'uid' */
    strncpy((char *) uid.arr, cliente, UNAME_LEN);
    uid.len = strlen((char *) uid.arr);

    /* Ejecutar la SQL de SELECCIÓN */
    EXEC SQL SELECT nombre, apellidos, edad
        INTO :tuplaCliente
        FROM Clientes
        WHERE id_cliente = :uid;

    /* Ver el resultado de la consulta de selección */
    if ( sqlca.sqlcode == 0 ) return SUCCESS;
    else if ( sqlca.sqlcode > 0 ) return NOT_EXISTS;
    else return ERROR;
}
```

Captura de Errores

```
void sql_error(char *msg)
{
    char err_msg[128];
    int buf_len, msg_len;

    EXEC SQL WHENEVER SQLERROR CONTINUE;

    /* Imprimir por salida estándar el mensaje msg (por defecto, "Error en Oracle:" */
    printf("\n%s\n", msg);

    /* Obtener el mensaje de error usando la función sqlglm1 */
    buf_len = sizeof (err_msg);
    sqlglm(err_msg, &buf_len, &msg_len);

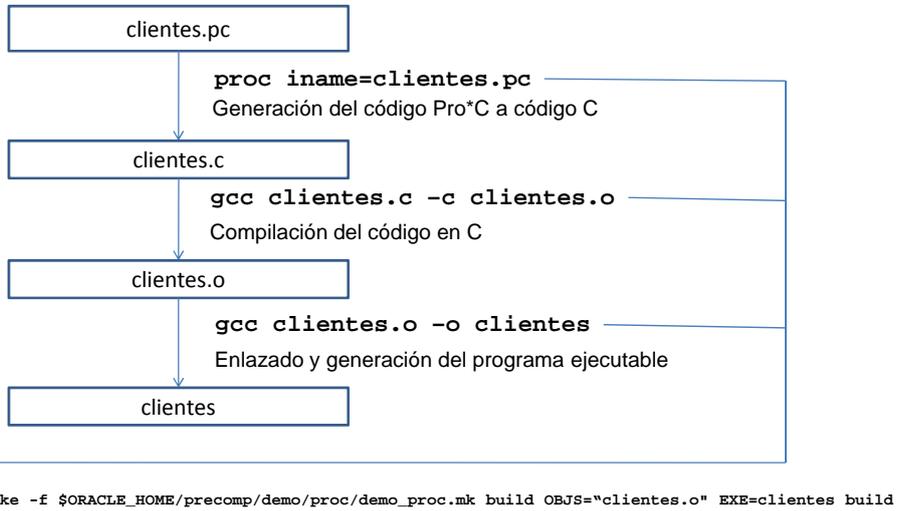
    /* Imprimir por salida estándar el mensaje de error */
    printf("%.*s\n", msg_len, err_msg);

    /* Devolver la Base de Datos al estado inmediatamente anterior de la ejecución de la
    * SQL */
    EXEC SQL ROLLBACK RELEASE;

    exit(1);
}
```

¹SQLCA puede almacenar hasta 70 mensajes de error en su variable sqlerrm. Para obtener el contenido de esta variable se utiliza la función sqlglm

Compilación



Utilización de makefiles

makefile

```
# Macros
CC      = gcc
PROC    = proc
PROG    = clientes
CFLAGS  = -DNDEBUG $(INCS)
INCS    = -I/usr/local/oracle/app/oracle/product/8.1.6/precomp/public
LDLFLAGS = -L/usr/local/oracle/app/oracle/product/8.1.6/precomp/public \
          /usr/local/oracle/app/oracle/product/8.1.6/lib/libclntsh.so

LIBS    =
SRCS_C  = clientes.c
SRCS_PC = clientes.pc
OBJS    = clientes.o

# Reglas
all:    $(PROG)

$(SRCS_C): $(SRCS_PC)
    @echo "Precompilando $(SRCS_PC) ..."
    $(PROC) $(SRCS_PC)

$(OBJS): $(SRCS_C)
    @echo "Generando código objeto para $(SRCS_C) ..."
    $(CC) $(CFLAGS) -c $<
```

Utilización de makefiles (cont.)

makefile (cont.)

```
$(PROG): $(OBJS)
    @echo "Compilando $(OBJS) y generando $(PROG)..."
    $(CC) $(OBJS) $(LDFLAGS) $(LIBS) -o $(PROG)
    @echo "$(PROG) creado."

clean:
    @echo "Borrando archivos objeto y ejecutable ..."
    rm -f core tags $(PROG) $(OBJS) $(SRCS_C)
    @echo "Finalizado."

# Dependencias
clientes.o: clientes.c
clientes.c: clientes.pc
```

Para compilar

```
> make
```

Para borrar el ejecutable y los código objeto

```
> make clean
```

Cuentas Hendrix y Oracle

- Los alumnos tienen la misma cuenta para Hendrix (hendrix.cps.unizar.es) y Oracle (vicious.den.cps.unizar.es)
- Para los nuevos usuarios (los que no tenían cuenta anteriormente), solicitar a Pedro Álvarez las cuentas y contraseñas
- Para los que han olvidado la contraseña, hablar con el administrador (J. A. Gutierrez)