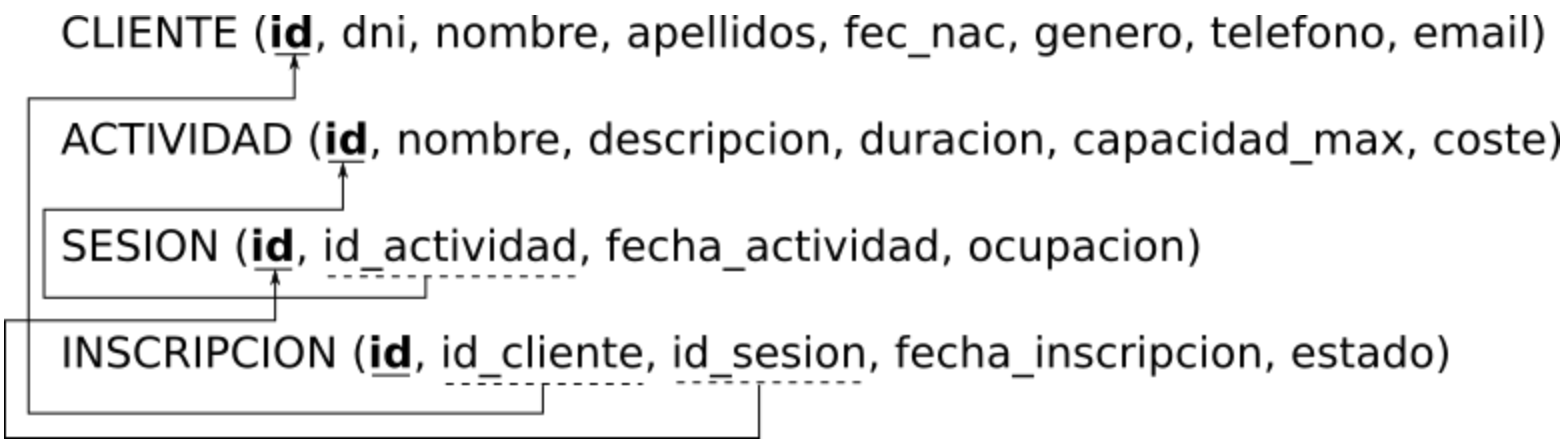
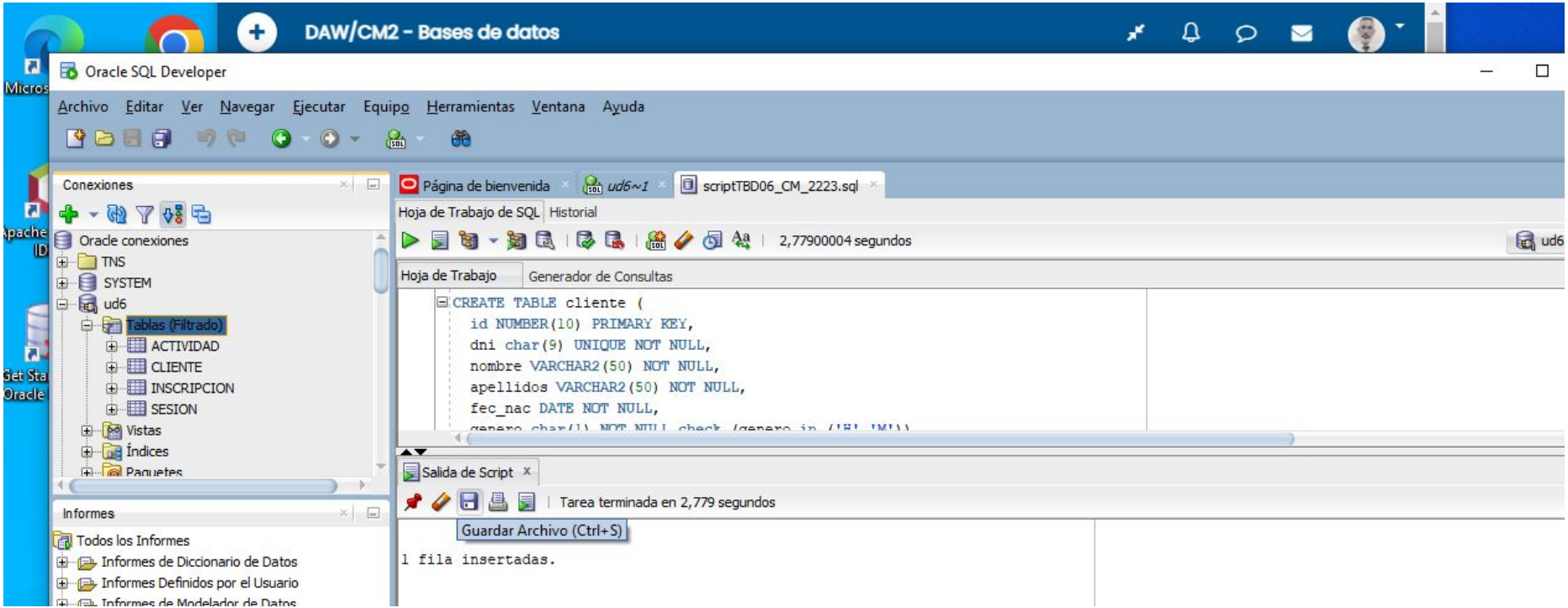
Programación en la Base de datos "Actividades deportivas"

Unidad: 6. Programación de bases de datos.

FRANCISCO JAVIER GUTIÉRREZ PÉREZ


CLIENTE (id, dni, nombre, apellidos, fec_nac, genero, telefono, email)
ACTIVIDAD (id, nombre, descripcion, duracion, capacidad_max, coste)
SESION (id, id_actividad, fecha_actividad, ocupacion)
INSCRIPCION (id, id_cliente, id_sesion, fecha_inscripcion, estado)


DAW/CM2 - Bases de datos
Oracle SQL Developer
Archivo Editar Ver Navegar Ejecutar Equipo Herramientas Ventana Ayuda
Conexiones
Oracle conexiones
TNS
SYSTEM
ud6
Tablas (Filtrado)
ACTIVIDAD
CLIENTE
INSCRIPCION
SESION
Vistas
Índices
Paquetes
Informes
Todos los Informes
Informes de Diccionario de Datos
Informes Definidos por el Usuario
Página de bienvenida
ud6~1
scriptTBD06_CM_2223.sql
Hoja de Trabajo de SQL
Historial
2,77900004 segundos
ud6
Hoja de Trabajo
Generador de Consultas
CREATE TABLE cliente (
id NUMBER(10) PRIMARY KEY,
dni char(9) UNIQUE NOT NULL,
nombre VARCHAR2(50) NOT NULL,
apellidos VARCHAR2(50) NOT NULL,
fec_nac DATE NOT NULL,
Salida de Script
Tarea terminada en 2,779 segundos
Guardar Archivo (Ctrl+S)
1 fila insertadas.

Actividad 1: (Procedimiento 1)

Crear un procedimiento llamado **actualizar_ocupacion** de forma que tenga como parámetros de entrada:

- identificador de una sesión

El procedimiento debe de actualizar el número de inscritos de una sesión, para ello debes realizar las siguientes acciones:

- Mostrar un mensaje por consola en el caso de que la sesión pasada por parámetros no exista en la base de datos
- En el caso de que la sesión exista, se deben contar las inscripciones realizadas por los clientes para esa sesión (sólo las que se encuentren en estado "Reservado") y actualizar en la base de datos la ocupación de la sesión.
- Una vez actualizada la ocupación, se mostrará un mensaje por consola, indicando el nombre de la actividad a la que pertenece dicha sesión, la fecha de la sesión, el código de la sesión, la ocupación actual y la ocupación máxima de la actividad. El formato de salida debe ser similar a las imágenes que se muestran en los siguientes ejemplos de llamada al procedimiento:

Nota: La ocupación máxima de cada sesión, viene indicada en la actividad a la que pertenece dicha sesión.

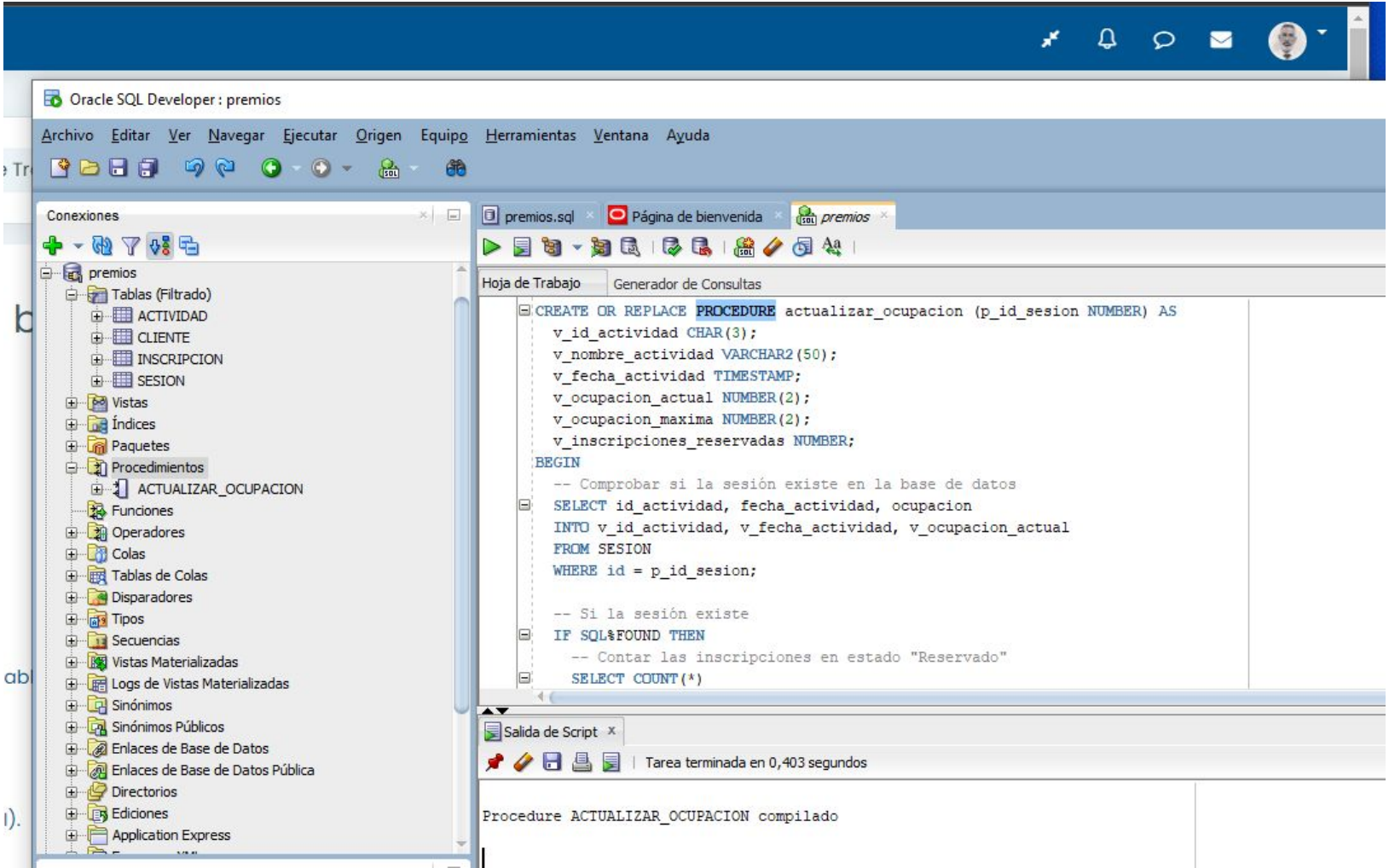
Ejemplo de llamadas al procedimiento:

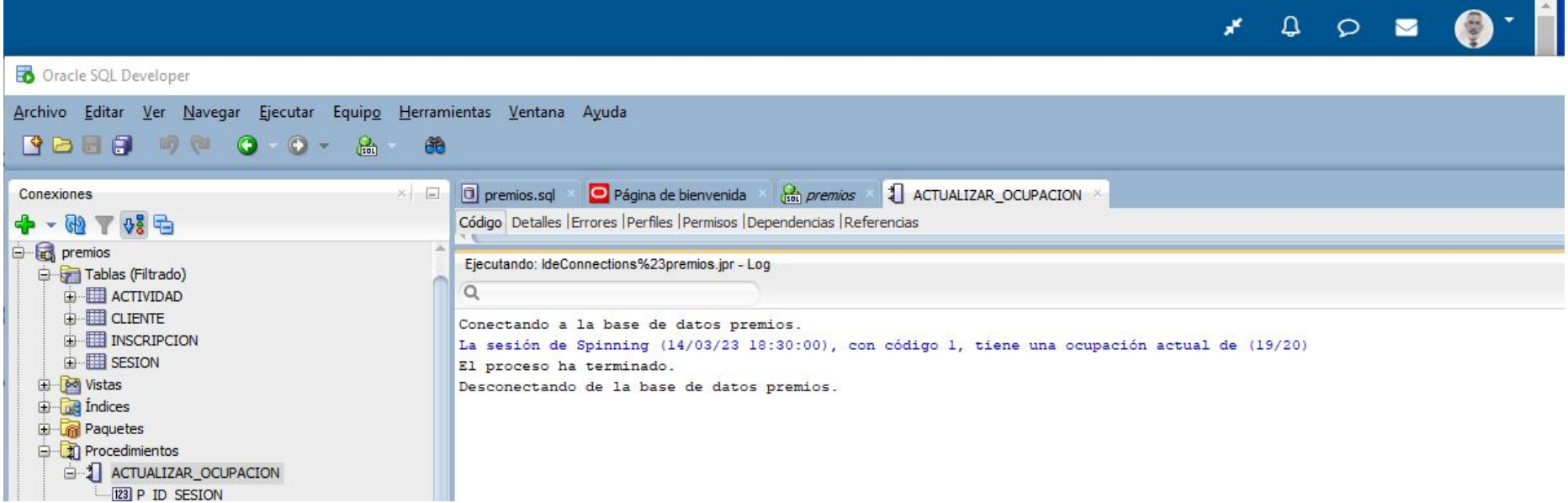
```
actualizar_ocupacion (1);  /* llamada al procedimiento para la sesión con identificador 1 */
```

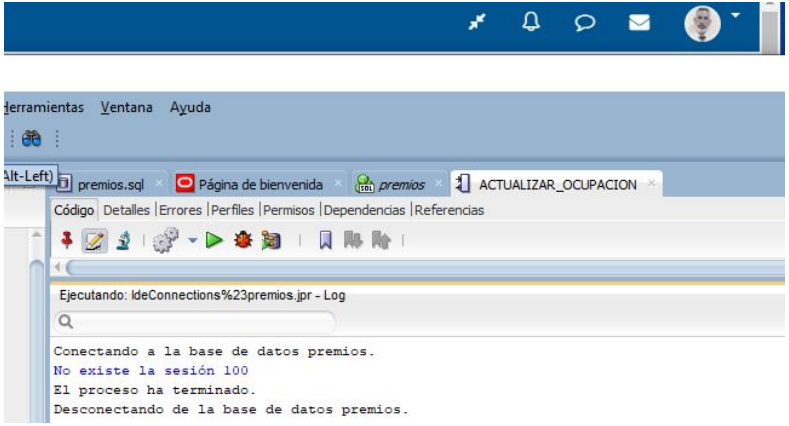
```
La sesión de Spinning(14/03/23 18:30:00,000000), con código 1, tiene una ocupación actual de (19/20)
```

```
actualizar_ocupacion (100);  /* llamada al procedimiento para la sesión con identificador 100 */
```

```
No existe la sesión con código 100
```


Oracle SQL Developer : premios
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones
premios
Tablas (Filtrado)
ACTIVIDAD
CLIENTE
INSCRIPCION
SESION
Vistas
Índices
Paquetes
Procedimientos
ACTUALIZAR_OCUPACION
Funciones
Operadores
Colas
Tablas de Colas
Disparadores
Tipos
Secuencias
Vistas Materializadas
Logs de Vistas Materializadas
Sinónimos
Sinónimos Públicos
Enlaces de Base de Datos
Enlaces de Base de Datos Pública
Directorios
Ediciones
Application Express
premios.sql
Página de bienvenida
premios
Hoja de Trabajo
Generador de Consultas
CREATE OR REPLACE PROCEDURE actualizar_ocupacion (p_id_sesion NUMBER) AS
v_id_actividad CHAR(3);
v_nombre_actividad VARCHAR2(50);
v_fecha_actividad TIMESTAMP;
v_ocupacion_actual NUMBER(2);
v_ocupacion_maxima NUMBER(2);
v_inscripciones_reservadas NUMBER;
BEGIN
-- Comprobar si la sesión existe en la base de datos
SELECT id_actividad, fecha_actividad, ocupacion
INTO v_id_actividad, v_fecha_actividad, v_ocupacion_actual
FROM SESION
WHERE id = p_id_sesion;
-- Si la sesión existe
IF SQL%FOUND THEN
-- Contar las inscripciones en estado "Reservado"
SELECT COUNT(*)
Salida de Script
Tarea terminada en 0,403 segundos
Procedure ACTUALIZAR_OCUPACION compilado


Oracle SQL Developer
Archivo Editar Ver Navegar Ejecutar Equipo Herramientas Ventana Ayuda
Conexiones
premios
Tablas (Filtrado)
ACTIVIDAD
CLIENTE
INSCRIPCION
SESION
Vistas
Índices
Paquetes
Procedimientos
ACTUALIZAR_OCUPACION
P_ID_SESION
premios.sql
Página de bienvenida
premios
ACTUALIZAR_OCUPACION
Código Detalles Errores Perfiles Permisos Dependencias Referencias
Ejecutando: IdeConnections%23premios.jpr - Log
Conectando a la base de datos premios.
La sesión de Spinning (14/03/23 18:30:00), con código 1, tiene una ocupación actual de (19/20)
El proceso ha terminado.
Desconectando de la base de datos premios.


Herramientas Ventana Ayuda
premios.sql
Página de bienvenida
premios
ACTUALIZAR_OCUPACION
Código Detalles Errores Perfiles Permisos Dependencias Referencias
Ejecutando: IdeConnections%23premios.jpr - Log
Conectando a la base de datos premios.
No existe la sesión 100
El proceso ha terminado.
Desconectando de la base de datos premios.

Actividad 2: (Función 2).

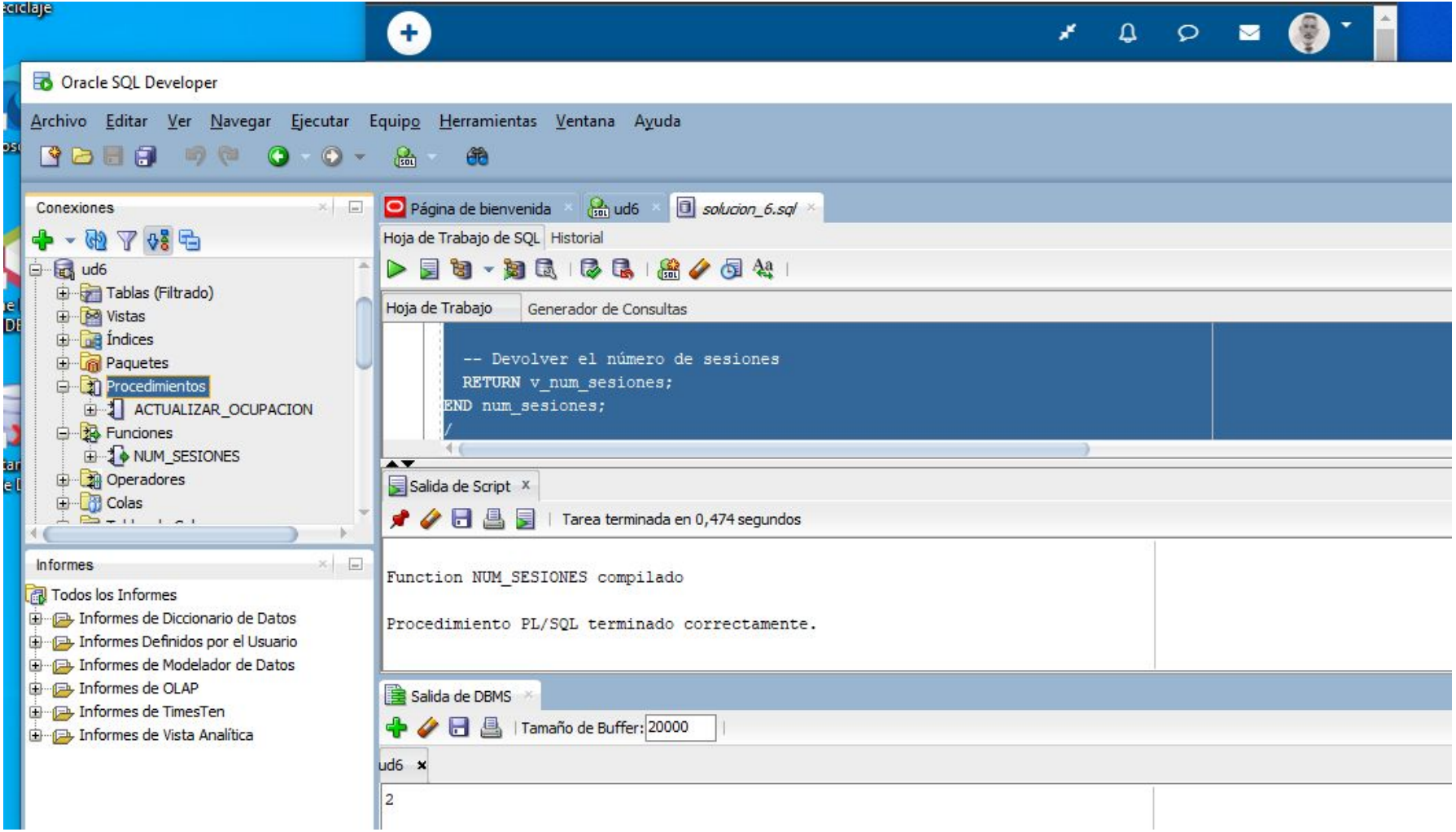
Crear una función llamada **num_sesiones**, donde a partir de un identificador de una actividad y un rango de fechas, devuelva el número de sesiones existentes de dicha actividad comprendidas entre la fecha de inicio y de fin. Parámetros de entrada a la función

- id_actividad
- fecha_inicio
- fecha_fin

Hay que tener en cuenta que los parámetros fecha_inicio y fecha_fin son opcionales y que el valor por defecto de ambos es la fecha actual del sistema,

Ejemplo de llamadas a la función:

> *dbms_output.put_line(**num_sesiones**('SPI', to_date('12/04/2023','DD/MM/YYYY'), to_date('12/05/2023','DD/MM/YYYY')));* --> *la función devuelve 2 sesiones comprendidas entre esas fechas*
> *dbms_output.put_line(**num_sesiones**('YOG', to_date('12/01/2023','DD/MM/YYYY')));* --> *la función devuelve 1 (teniendo en cuenta que la fecha del sistema al realizar la prueba ha sido el 15/03/2023)*


Oracle SQL Developer
Archivo Editar Ver Navegar Ejecutar Equipo Herramientas Ventana Ayuda
Conexiones
ud6
Tablas (Filtrado)
Vistas
Índices
Paquetes
Procedimientos
ACTUALIZAR_OCUPACION
Funciones
NUM_SESIONES
Operadores
Colas
Informes
Todos los Informes
Informes de Diccionario de Datos
Informes Definidos por el Usuario
Informes de Modelador de Datos
Informes de OLAP
Informes de TimesTen
Informes de Vista Analítica
Página de bienvenida
ud6
solucion_6.sql
Hoja de Trabajo de SQL
Historial
Hoja de Trabajo
Generador de Consultas
-- Devolver el número de sesiones
RETURN v_num_sesiones;
END num_sesiones;
/
Salida de Script
Tarea terminada en 0,474 segundos
Function NUM_SESIONES compilado
Procedimiento PL/SQL terminado correctamente.
Salida de DBMS
Tamaño de Buffer: 20000
ud6
2

Actividad 3: (Procedimiento 2)

Crear un procedimiento llamado **info_cliente** de forma que pasándole como parámetros de entrada:

- dni de un cliente
  **se pasa el dni de un cliente no el identificador del cliente*

El procedimiento debe realizar una serie de comprobaciones y mostrar por pantalla la siguiente información

- El nombre y apellidos del cliente a través del siguiente mensaje: 'El cliente "*nombre apellidos*" ha realizado las siguientes actividades'
- En el caso de que el cliente no exista, se debe indicar con otro mensaje a través de la consola.

Además se mostrará por pantalla un listado de las actividades que ha realizado el cliente, la fecha de las actividades y el coste de cada actividad. Para ello hay que tener en cuenta:

- Sólo se mostrarán las actividades que el cliente fue inscrito con el estado "Reservado"
- El listado aparecerá ordenado por fecha de actividad desde la más reciente a la más antigua.
- La fecha de la actividad se mostrará con el formato "día/mes/año hora:minutos"
- El coste se mostrará con el símbolo del €
- Hacer uso de las funciones LPAD y RPAD para mostrar los datos correctamente posicionados tal y como se muestran en las siguientes imágenes

Finalmente el procedimiento mostrará el coste total del cliente.

- El coste total del cliente se calcula sumando todas las actividades del cliente en estado "Reservado"

Para la resolución de esta actividad tienes que hacer uso de cursores.

Ejemplo de llamadas al procedimiento y mensaje mostrado por consola:

info_cliente('50984975R'); / llamada al procedimiento para el dni 50984975R*/*

```
El cliente "Juan González Sánchez" ha realizado las siguientes actividades

Actividad   Fecha actividad     Coste
--------------------------------------
Spinning    12/04/2023 18:30     6€
Pilates     11/04/2023 18:00    10€
Yoga        14/03/2023 20:00     7€
Spinning    14/03/2023 18:30     6€
--------------------------------------
Total : 29€
```

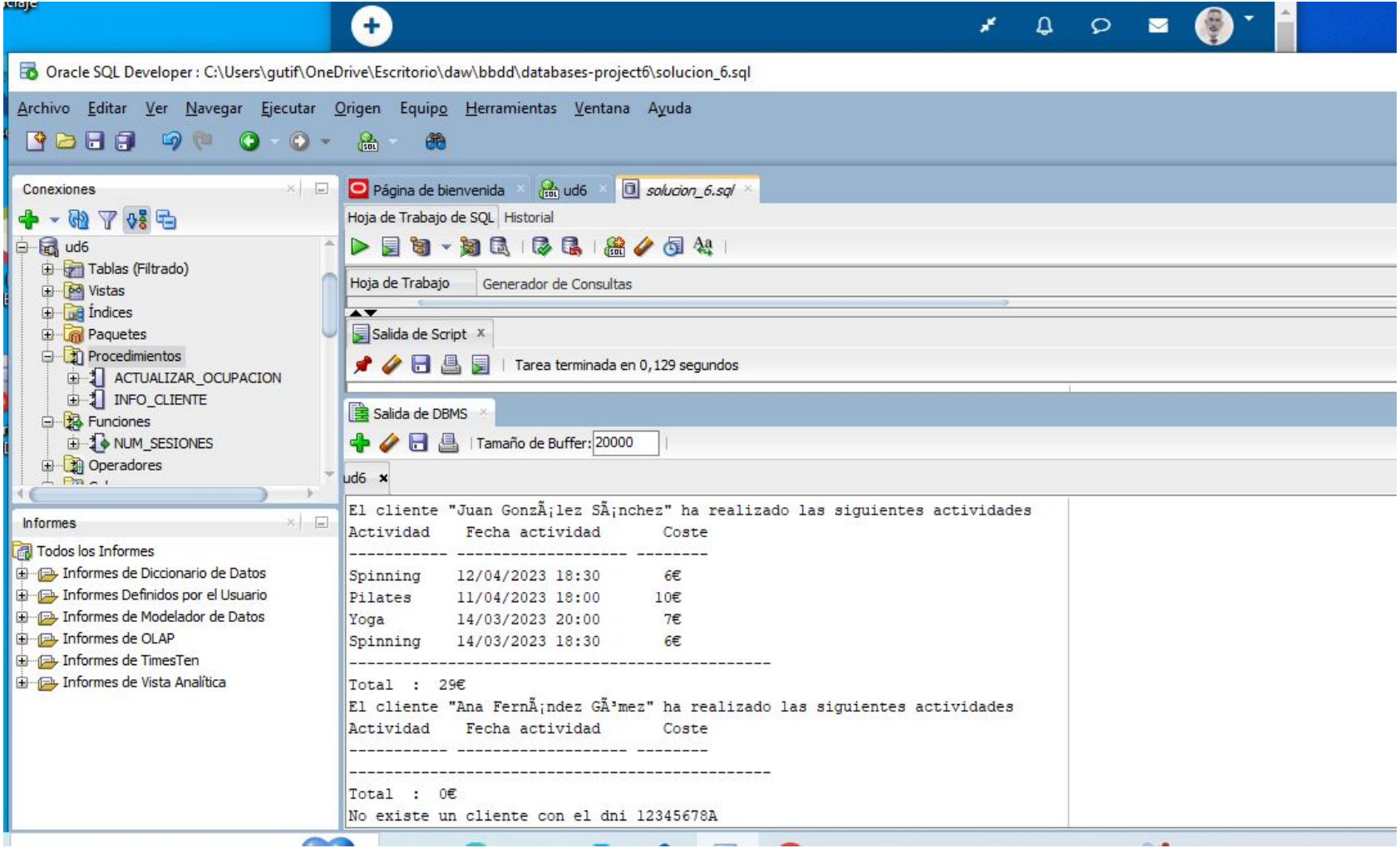
info_cliente('06349006V'); / llamada al procedimiento para el dni 06349006V*/*

```
El cliente "Ana Fernández Gómez" ha realizado las siguientes actividades

Actividad   Fecha actividad     Coste
--------------------------------------
--------------------------------------
Total : 0€
```

info_cliente('12345678A'); / llamada al procedimiento para el dni 12345678A*/*

```
No existe un cliente con el dni 12345678A
```


Oracle SQL Developer : C:\Users\gutif\OneDrive\Escritorio\daw\bbdd\databases-project6\solucion_6.sql
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones
ud6
Tablas (Filtrado)
Vistas
Índices
Paquetes
Procedimientos
ACTUALIZAR_OCUPACION
INFO_CLIENTE
Funciones
NUM_SESIONES
Operadores
Informes
Todos los Informes
Informes de Diccionario de Datos
Informes Definidos por el Usuario
Informes de Modelador de Datos
Informes de OLAP
Informes de TimesTen
Informes de Vista Analítica
Página de bienvenida
ud6
solucion_6.sql
Hoja de Trabajo de SQL
Historial
Hoja de Trabajo
Generador de Consultas
Salida de Script
Tarea terminada en 0,129 segundos
Salida de DBMS
Tamaño de Buffer: 20000
ud6
El cliente "Juan GonzÃ¡lez SÃ¡nchez" ha realizado las siguientes actividades
Actividad Fecha actividad Coste
Spinning 12/04/2023 18:30 6€
Pilates 11/04/2023 18:00 10€
Yoga 14/03/2023 20:00 7€
Spinning 14/03/2023 18:30 6€
Total : 29€
El cliente "Ana FernÃ¡ndez GÃ³mez" ha realizado las siguientes actividades
Actividad Fecha actividad Coste
Total : 0€
No existe un cliente con el dni 12345678A

Actividad 4: (Trigger 4)

Crear un **trigger** o **disparador** llamado **insertar_inscripcion** de forma que cuando se vaya a insertar una inscripción de un cliente en una sesión, antes de grabarlo se hagan una serie de acciones que se detallan a continuación:

- Comprobaremos que la inscripción que se va a insertar va a estar en estado "Reservado"
- Comprobaremos que el cliente no estaba inscrito previamente en dicha sesión o bien tenía alguna inscripción en estado "Cancelado"

Si se cumplen, se debe permitir la inscripción del cliente en la sesión, pero hay que tener en cuenta:

- Si tenía una inscripción en estado "Cancelada", se debe eliminar previamente.

Si no se cumplen ambas condiciones no se debe insertar el registro. Debes lanzar diferentes excepciones con mensajes que detallen qué condiciones incumple para no poder insertar el registro.

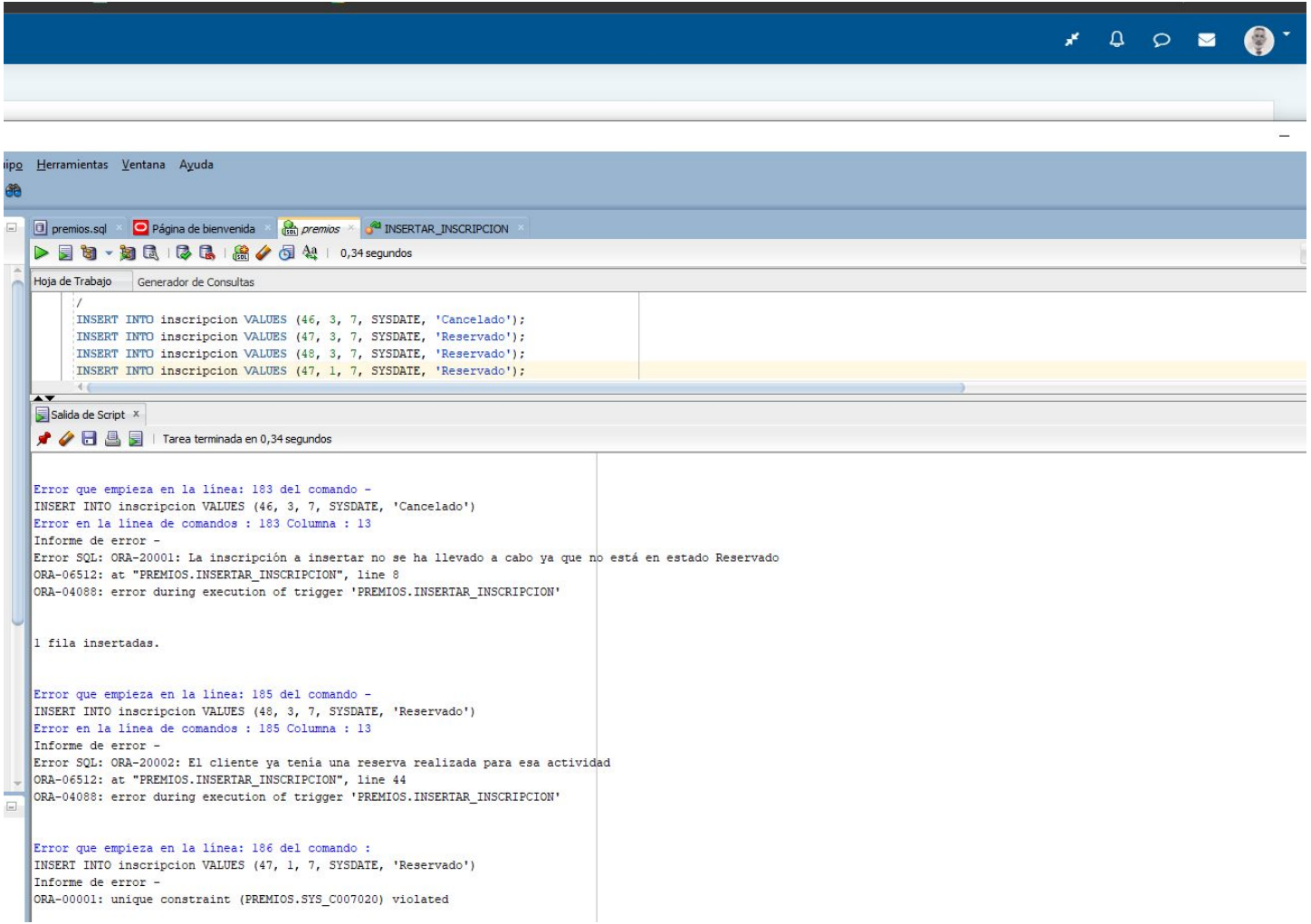
Ejemplo de inserción:

Ejemplo de inserción que no se lleva a cabo por estar su estado en "Cancelado"
INSERT INTO inscripcion VALUES (46, 3, 7, SYSDATE, 'Cancelado'); *-- Se muestra el mensaje "La inscripción a insertar no se ha llevado a cabo ya que no está en estado Reservado"*

Ejemplo de inserción llevada a cabo ya que cumple todos los requisitos
INSERT INTO inscripcion VALUES (47, 3, 7, SYSDATE, 'Reservado'); *-- se muestra un mensaje "Inscripción insertada"*

Ejemplo de inserción que ya existía previamente. El cliente ya estaba inscrito a la sesión
INSERT INTO inscripcion VALUES (48, 3, 7, SYSDATE, 'Reservado'); *--Se muestra el mensaje: "El cliente ya tenía una reserva realizada para esa actividad"*

Ejemplo de inserción que ya existía previamente en estado Cancelada. Se procede a borrar la antigua inscripción y a insertar la nueva
INSERT INTO inscripcion VALUES (47, 1, 7, SYSDATE, 'Reservado'); *-- Se muestra el mensaje: "El cliente tenía una reserva en estado Cancelado. Se elimina la reserva anterior. Inscripción realizada."*


Herramientas Ventana Ayuda
premios.sql
Página de bienvenida
premios
INSERTAR_INSCRIPCION
0,34 segundos
Hoja de Trabajo
Generador de Consultas
/
INSERT INTO inscripcion VALUES (46, 3, 7, SYSDATE, 'Cancelado');
INSERT INTO inscripcion VALUES (47, 3, 7, SYSDATE, 'Reservado');
INSERT INTO inscripcion VALUES (48, 3, 7, SYSDATE, 'Reservado');
INSERT INTO inscripcion VALUES (47, 1, 7, SYSDATE, 'Reservado');
Salida de Script
Tarea terminada en 0,34 segundos
Error que empieza en la línea: 183 del comando -
INSERT INTO inscripcion VALUES (46, 3, 7, SYSDATE, 'Cancelado')
Error en la línea de comandos : 183 Columna : 13
Informe de error -
Error SQL: ORA-20001: La inscripción a insertar no se ha llevado a cabo ya que no está en estado Reservado
ORA-06512: at "PREMIOS.INSERTAR_INSCRIPCION", line 8
ORA-04088: error during execution of trigger 'PREMIOS.INSERTAR_INSCRIPCION'
1 fila insertadas.
Error que empieza en la línea: 185 del comando -
INSERT INTO inscripcion VALUES (48, 3, 7, SYSDATE, 'Reservado')
Error en la línea de comandos : 185 Columna : 13
Informe de error -
Error SQL: ORA-20002: El cliente ya tenía una reserva realizada para esa actividad
ORA-06512: at "PREMIOS.INSERTAR_INSCRIPCION", line 44
ORA-04088: error during execution of trigger 'PREMIOS.INSERTAR_INSCRIPCION'
Error que empieza en la línea: 186 del comando :
INSERT INTO inscripcion VALUES (47, 1, 7, SYSDATE, 'Reservado')
Informe de error -
ORA-00001: unique constraint (PREMIOS.SYS_C007020) violated