ÍNDICE

Contenido

Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶

Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶
GOBIERNO DE ESPAÑA MINISTERIO DE HACIENDA

Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶

Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶

CURSORES AVANZADOS.

Cuando nos enfrentamos al diseño de un cursor, tenemos que tener en cuenta el tipo de bucle que vamos a utilizar para extraer los datos y consecuentemente, para recorrer las filas que reporte la consulta.

Disponemos de 3 tipos de bucles que se puede utilizar con un cursor:

- Bucles simples (LOOP…END LOOP).
- Bucles WHILE.
- Bucles FOR.

Bucles simples (LOOP … END LOOP)

Implican el proceso completo de un cursor explícito: Apertura, recorrido y cierre.

El proceso de recorrido por los valores que devuelve el cursor se realizará implementando un bucle LOOP ... END LOOP.

Por ejemplo:

```
DECLARE

      V_nombre          estudiantes%TYPE;
      v_apellido        estudiantes%TYPE;
      CURSOR c_nombres IS
            SELECT nombre, apellido from estudiantes;
```

```
BEGIN
      OPEN c_nombres;
      LOOP
            FETCH c_nombres INTO v_nombre, v_apellido;
            EXIT WHEN c_nombres%NOTFOUND;

            /* Tratamiento del registro */

      END LOOP;
      CLOSE c_nombres;
END;
```

Bucles WHILE

Implican el proceso completo de un cursor explícito: Apertura, recorrido y cierre.

El proceso de recorrido por los valores que devuelve el cursor se realizará implementando un bucle LOOP ... END LOOP.

Por ejemplo:

```
DECLARE
      v_nombre   estudiantes%TYPE;
      v_apellido estudiantes%TYPE;
      CURSOR c_nombres IS
            SELECT nombre, apellido from estudiantes;
BEGIN
      OPEN c_nombres;
      FETCH c_nombres INTO v_nombre, v_apellido;
      WHILE c_nombres%FOUND LOOP

            /* Tratamiento del registro */
            FETCH c_nombres INTO v_nombre, v_apellido;
      END LOOP;
      CLOSE c_nombres;
END;
```

Bucles FOR

Este tipo de bucles no hace uso del proceso de un cursor explícito pero tampoco se puede considerar un cursor implícito, dado que a los que se realizan con bucles FOR se les nomina previamente (característica de los cursores explícitos).

Así pues es un tipo de cursor que comparte características de ambos. En cuanto a las características que le puedan asimilar a un cursor implícito, se encuentra que la apertura, cierre y recorrido del cursor se realizan de forma implícita por el bucle, sin necesidad de expresar órdenes para realizar estas acciones.

El proceso de recorrido por los valores que devuelve el cursor se realizará implementando un bucle FOR LOOP ... END LOOP.

Por ejemplo:

```
DECLARE
      v_nombre   estudiantes%TYPE;
      v_apellido estudiantes%TYPE;
      CURSOR c_nombres IS
            SELECT nombre, apellido from estudiantes;
BEGIN
      /*
       Al hacer el FOR se abre el cursor y se recupera la
primera fila si se puede asignándolo a la variable implícita
que no hay que declarar v_nombres.
      */

      FOR v_nombres IN c_nombres LOOP

            -- Tratamiento que se vaya a realizar sobre los
            -- registros.

            /*  Los  movimientos  entre  registros  son
automáticos, no se necesita el fetch haciéndose aquí una
comprobación implícita de c_nombres%NOTFOUND. */

      END LOOP;
```

```
      -- Al hacer el end loop se cierra implícitamente el
      -- cursor abierto.
END;
```

CURSORES SELECT FOR UPDATE

Hasta ahora habíamos visto cursores que consultaban un conjunto de registros y que posteriormente se operaba sobre los valores de esas filas para realizar otras operaciones.

Los cursores SELECT FOR UPDATE tienen como misión la actualización de las propias filas que retorna el cursor.

Poseen 2 partes diferenciadas:

- FOR UPDATE
- WHERE CURRENT OF

For update

Es la última cláusula de la instrucción SELECT. Aparecerá al final de la instrucción tras la cláusula ORDER BY (si es que se utiliza esta última para hacer el SELECT).

La sintaxis es:

```
SELECT …
FROM …
FOR UPDATE [OF columna1[, …columnaX]] [NOWAIT];
```

Se puede indicar 1, varias o ninguna columna para actualizar después del OF. En caso de hacerlo únicamente se podrán actualizar en la tabla, las columnas que se hayan incluido después de FOR UPDATE OF.

Si no se indica la cláusula OF, se podrá actualizar cualquier columna que retorne el cursor.

Cuando se realiza un SELECT FOR UPDATE se bloquean todas las filas que reporta el cursor, no siendo accesibles para ningún otro proceso que realice una operación sobre las filas bloqueadas de la tabla.

Todo intento de modificar dichas filas quedará congelado hasta que el proceso que ha abierto el cursor para actualización ejecute un COMMIT.

La cláusula NOWAIT impide que otro SELECT FOR UPDATE se quede esperando al COMMIT, inmediatamente presenta un error que indica que las filas están bloqueadas y se ha decidido no esperar hasta que sean desbloqueadas.

EJEMPLO

```
DECLARE
      CURSOR c_estudiantes IS
            SELECT * FROM estudiantes
            FOR UPDATE OF nombre, apellidos;
```

En este ejemplo hemos definido un cursor para actualizar la tabla ESTUDIANTES. Únicamente permitiremos actualizar las columnas NOMBRE y APELLIDOS de dicha tabla.

EJEMPLO

```
DECLARE
      CURSOR c_estudiantes IS
            SELECT curso FROM estudiantes
            WHERE curso = 10
            FOR UPDATE;
```

En este ejemplo hemos definido un cursor para actualizar la tabla ESTUDIANTES. En este caso si que permitimos actualizar cualquier columna de la tabla.

Where current of

Esta cláusula sólo se utiliza en una sentencia UPDATE para indicar que existe un cursor SELECT … FOR UPDATE desde el que se realizará la operación de modificación y únicamente sobre las filas que ha retornado dicho cursor.

La sintaxis es:

```
UPDATE tabla_del_cursor
SET
      Columna1 … columna X
WHERE CURRENT OF nombre_cursor;
```

EJEMPLO

```
DECLARE
     CURSOR c_estudiantes IS
          SELECT * FROM estudiantes
          FOR UPDATE OF nombre, apellidos;
BEGIN
     FOR v_estudiante IN c_estudiantes LOOP
          UPDATE estudiantes
          SET
               Nombre = 'X-'||nombre
          WHERE CURRENT OF c_estudiantes;
     END LOOP;
     COMMIT;
END;
```

En este ejemplo hemos definido un cursor para actualizar la tabla ESTUDIANTES. Únicamente permitiremos actualizar las columnas NOMBRE y APELLIDOS de dicha tabla.

Durante el bucle de recorrido del cursor se actualiza cada fila retornada, actualizando la columna NOMBRE, introduciéndole la cadena 'X-' delante del contenido que ya tuviese.

EJEMPLO

```
DECLARE
      CURSOR c_estudiantes IS
            SELECT curso FROM estudiantes
            WHERE curso = 10
            FOR UPDATE;
BEGIN
      FOR v_estudiante IN c_estudiantes LOOP
            UPDATE estudiantes
            SET
                  id_matricula = id_matricula||'_2001'
            WHERE CURRENT OF c_estudiantes;
      END LOOP;
      COMMIT;
END;
```

En este ejemplo hemos definido un cursor para actualizar la tabla ESTUDIANTES. En este caso sí que permitimos actualizar cualquier columna de la tabla y durante el recorrido de las filas que retorna el cursor, se actualiza la columna MATRICULA introduciendo al contenido que ya tuviese la cadena '_2001'.

OBJETOS

Los objetos son una de las principales características que ya introdujo la versión de Oracle 8 con respecto a sus versiones anteriores. De esta forma PL/SQL se convirtió en un lenguaje no solo orientado a la programación sino también a los objetos, característica esta última de los lenguajes más extendidos, como el lenguaje C entre otros.

Bases de la programación orientada a objetos

¿Por qué escribimos aplicaciones informáticas? Una posible respuesta es: para modelar el mundo real.

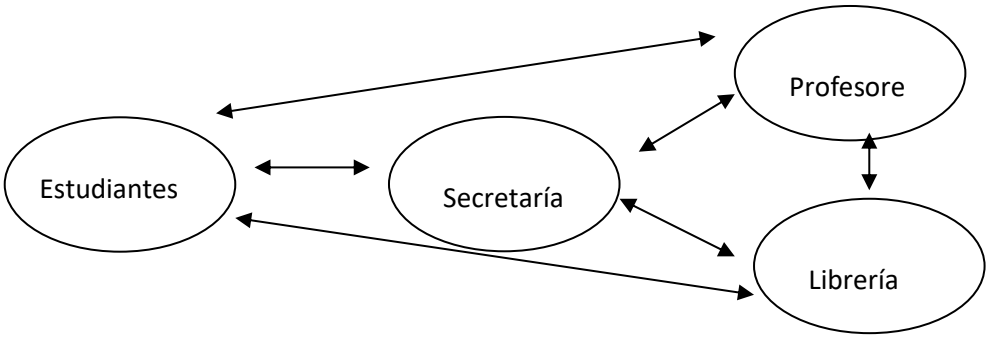
Un sistema software se diseña para simular los objetos que hay en el mundo y las interacciones existentes entre ellos.

Una vez que se han modelado los objetos y sus interacciones, la aplicación se puede utilizar para ver cómo evolucionan, y automatizar los procesos implicados.

Para poder entender mejor la programación orientada a objetos, consideremos el siguiente ejemplo:

- Consideremos una universidad. ¿Cuáles son las entidades de este mundo? Tenemos estudiantes, que se comunican con la secretaría para contratar los cursos.
- La secretaría informa a los profesores que los estudiantes se han apuntado a los cursos.
- Los profesores se comunican con los estudiantes durante las clases y cuando asignan los niveles.
- Los profesores deben informar a las personas que rigen la librería de la universidad qué libros desean para sus clases, para poder ponerlos a disposición de los estudiantes.

Este modelo quedaría ilustrado de la forma que aparece en la siguiente figura.

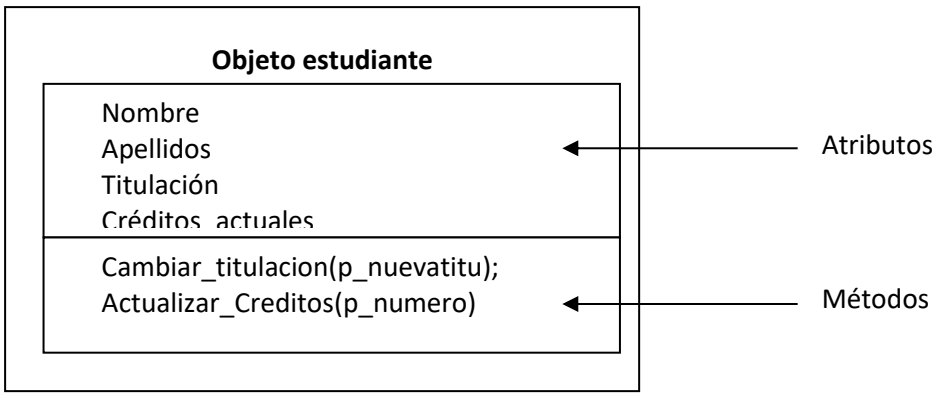


El modelo orientado a objetos implementa directamente este modelo en una aplicación informática, con las siguientes premisas:

- Cada una de las entidades se representa mediante un objeto del sistema.
- Un objeto representa los atributos de la entidad del mundo real, y las operaciones que actúan sobre dichos atributos.

Por ejemplo y continuando con el ejemplo de la universidad antes descrito:

Consideremos el objeto estudiante, que se representa en la figura que se muestra a continuación.



Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶

La interpretación que hacemos de este objeto es que un estudiante tiene atributos tales como el nombre, los apellidos, su titulación y los créditos actuales.

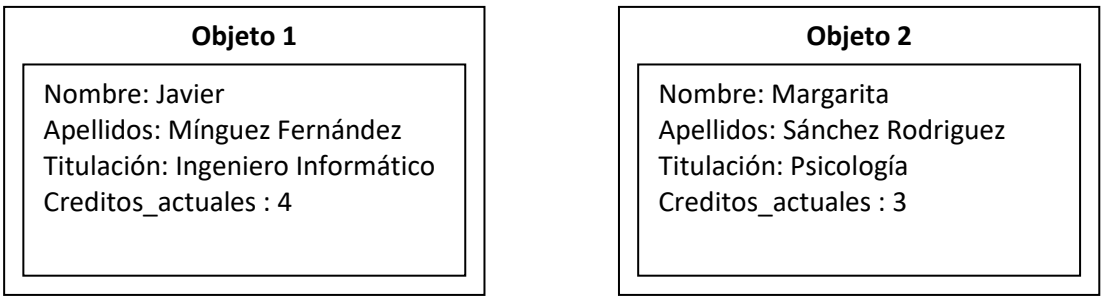
Además para poder interactuar con este objeto también se incluyen las operaciones que actúan sobre estos atributos tales como el cambio de titulación y la adición de créditos. Esto es lo que se llama los métodos de un objeto.

Objetos e instancias de los objetos

Es importante destacar la diferencia entre un tipo de objeto y una instancia de dicho tipo. En un entorno dado sólo puede haber un tipo de objeto, aunque pueden existir muchas instancias de él.

Una instancia de objeto es similar a una variable: cada instancia tiene su propia memoria y por tanto su propia copia de los atributos.

La figura que se muestra a continuación presente 2 instancias del objeto tipo estudiante.



Bases de datos objeto-relacionales

Como se ha comentado al comienzo de este capítulo, existen muchos lenguajes de programación orientados a objetos, entre los que se incluyen: C, C++ y Java.

Estos lenguajes permiten definir objetos y manipularlos. Sin embargo de lo que están faltos estos lenguajes es de características como la persistencia, la capacidad para almacenar y recuperar objetos de forma segura y la consistente.

Aquí es donde entran las bases de datos objeto-relacionales como Oracle, la cual está diseñada para almacenar y recuperar datos de objetos igual que sucede con los datos relacionales, utilizando el SQL como método estándar de comunicación con la base de datos.

En una base de datos objeto-relacional se pueden usar los lenguajes SQL y PL/SQL para manipular datos de objetos y datos relacionales.

Oracle12c también proporciona las ventajas de un control de transacciones consistente, de la realización de copias de seguridad y recuperación, de un rendimiento excelente en operaciones de consulta y de las funcionalidades de bloqueo, concurrencia y de escalabilidad.

Combinando los objetos y el modelo relacional tenemos lo mejor de ambos mundos: la potencia y confiabilidad de una base de datos relacional junto con la flexibilidad y capacidades de modelización de los objetos.

Definición de los tipos de objetos

La definición de un objeto en Oracle se realiza de una forma muy similar a la definición de un paquete de base de datos, donde aparece una especificación y cuerpo.

En este apartado aprenderemos a realizar las siguientes operaciones:

- Definición de la especificación de un objeto.
- Declaración, inicialización y asignación de valores a un objeto.
- Especificación de métodos.
- Definición del cuerpo de un objeto.

Especificación de un objeto

La sintaxis es:

```
CREATE [OR REPLACE] TYPE [esquema.]nombre_tipo
AS OBJECT
(nombre_atributo          tipo
[,nombre2                 tipo] …
| [{MAP | ORDER} MEMBER especificación_función]
| [MEMBER {especif_funcion | especif_procedimiento}
| [,MEMBER … ]);
```

Hay varios puntos a destacar sobre los tipos de objetos

1. La orden CREATE TYPE es una orden DDL. Por tanto no se puede utilizar directamente en un bloque PL/SQL.
2. Es necesario tener el privilegio del sistema CREATE TYPE que es parte del papel RESOURCE, para poder crear un tipo objeto.
3. Los tipos objeto se crean como objetos del diccionario de datos. En consecuencia se crean en el esquema actual, a menos que se especifique un esquema diferente en la orden CREATE TYPE ... AS OBJECT.
4. Los atributos del nuevo tipo creado se especifican de forma similar a los campos de un registro PL/SQL o las columnas de una tabla en la orden CREATE TABLE.
5. A diferencia de los campos de registro, los atributos de un tipo de objeto no pueden restringirse para que tomen el valor NOT NULL, ni tampoco pueden inicializarse con un valor predeterminado.
6. Al igual que un registro PL/SQL, puede referenciar los atributos de un objeto utilizando la notación de punto.

Existen también varias restricciones sobre el tipo de datos de los atributos del objeto. En concreto los atributos de los objetos pueden ser de cualquier tipo de datos Oracle excepto:

- LONG o LONG RAW. Sin embargo pueden ser de tipo LOB.
- Cualquier tipo de lenguaje nacional tal como NCHAR, NVARCHAR2 o NCLOB.
- ROWID.
- Los tipos disponible sólo en PL/SQL pero no en la base de datos. Entre estos se incluyen BINARY_INTEGER, BOOLEAN, PLS_INTEGER, RECORD y REF CURSOR.

- Un tipo definido con %TYPE o %ROWTYPE.
- Tipos definidos dentro de un paquete PL/SQL.

EJEMPLO

```
CREATE OR REPLACE TYPE estudiante AS OBJECT
(id         NUMBER(5),
Nombre      VARCHAR2(20)
);
```

Declaración, inicialización y asignación de valores a un objeto

Al igual que cualquier otra variable PL/SQL un objeto se declara simplemente incluyéndolo en la sección declarativa de un bloque.

Por ejemplo, para declarar una variable de tipo objeto podemos utilizar la siguiente estructura:

```
DECLARE
      V_estudiante  estudiante;
```

Para declarar y a la vez inicializar una variable de tipo objeto podemos utilizar la siguiente estructura:

```
DECLARE
      V_estudiante  estudiante := estudiante(1000,'Pepe');
```

Si por el contrario lo que queremos es hacer una asignación de valores a una variable de tipo objeto previamente declarada, se utilizará la siguiente estructura (por ejemplo con la variable declarada anteriormente):

```
DECLARE
      V_estudiante  estudiante := estudiante(1000,'Pepe');
BEGIN
      V_estudiante.nombre := 'JUAN';
END;
```

Si un objeto no es inicializado entonces el objeto tendrá valor nulo (NULL) y sus atributos no estarán disponibles. Si por ejemplo queremos evaluar si un objeto está o no nulo, podemos utilizar la siguiente estructura:

```
DECLARE
      V_estudiante  estudiante;
BEGIN
      IF v_estudiante IS NULL THEN …
      END IF;
END;
```

Especificación de métodos

Para definir los métodos asociados a un objeto se utiliza la siguiente sintaxis:

```
MEMBER {FUNCTION | PROCEDURE} nombre
```

EJEMPLO

```
CREATE OR REPLACE TYPE estudiante AS OBJECT
(id         NUMBER(5),
 nombre     VARCHAR2(20),
 apellidos VARCHAR2(20),
 nota       NUMBER(2),
 MEMBER FUNCTION nombre_ape RETURN VARCHAR2,
 MEMBER PROCEDURE cambio_nota (nueva IN NUMBER)
);
```

Como se puede observar en el ejemplo, la definición de los métodos asociados a un objeto siempre se realiza después de la definición de atributos del mismo.

Cuerpo de un objeto

La definición del cuerpo de un objeto se realiza con la siguiente sintaxis:

```
CREATE [OR REPLACE] TYPE [esquema.]nombre
BODY AS
[{MAP | ORDER} MEMBER declaración_función;]
| [MEMBER {declaración_procedimiento | declaración_función};
   …
END;
```

EJEMPLO

```
CREATE OR REPLACE TYPE BODY estudiante AS
      MEMBER FUNCTION nombre_ape RETURN VARCHAR2 IS
      BEGIN
            RETURN nombre||' '||apellidos;
      END;
      MEMBER PROCEDURE cambio_nota (nueva IN NUMBER) IS
      BEGIN
            Nota := nueva;
      END;

END;
```

Llamada a un método

Para invocar a un método de un objeto hay que utilizar la siguiente sintaxis:

nombre_objeto.metodo

EJEMPLO

```
DECLARE
      V_estudiante  estudiante := estudiante (1000, 'PEPE',
'Rodriguez', 3);

BEGIN
      V_estuadiante.cambio_nota(10);
      DBMS_OUTPUT.PUT_LINE(v_estudiante.nombre_ape);
END;
```

Borrar un objeto

Sintaxis:

```
DROP TYPE nombre [FORCE]
```

Cuando se utiliza la sentencia de borrado de un objeto junto a la cláusula FORCE se está indicando a Oracle que elimine el tipo de objeto cuyo nombre se ha indicado, aunque puedan existir dependencias de el.

En caso de que no se indique la cláusula FORCE y existir dependencias del objeto el SGBD de Oracle no permitirá llevar a cabo el borrado del objeto.

Modificar un objeto

Sintaxis:

```
ALTER TYPE nombre COMPILE [SPECIFICATIONS | BODY]
```

Durante la modificación de un objeto sólo se permite la recompilación de una de las dos partes de las que consta un objeto: especificación o cuerpo.

No se permite cambiar el contenido de un objeto con esta cláusula, para ello habría que reescribir el comando de creación del objeto.

Como alternativa a la creación de nuevo de la especificación de un objeto, se admite el comando ALTER para reemplazar un objeto por otro, con la siguiente sintaxis:

```
ALTER TYPE nombre REPLACE AS OBJETCT (espec_tipo_objeto)
```

Creación de tablas de objetos

Al igual que está permita la creación de tabla relacionales, también está permitida la creación de tablas de objetos, con la sintaxis que se indica a continuación:

```
CREATE TABLE nombre OF objeto;
```

Inserción de valores en una tabla de objetos

Cuando necesitemos insertar valores en una tabla de objetos, utilizaremos la siguiente sintaxis:

```
INSERT INTO tabla VALUES (objeto(valor1,….,valorX));
```

Por ejemplo:

```
INSERT INTO tab_estudiante VALUES
(estudiante(2000, 'JOSE', 'FRANCISCO', 4));
```

CERTIFICACIONES DE ORACLE

Las certificaciones de Oracle están reconocidas por la industria, y son la mejor carta de presentación para ayudarle a tener éxito en su carrera profesional dentro del sector de las Tecnologías de la Información.

Las certificaciones de Oracle son un certificado que prueba la educación y experiencia recibida en los diversos productos de Oracle, y pueden acelerar su desarrollo profesional, mejorar su productividad y mejorar su credibilidad.

Certificaciones de oracle disponibles

Toda la información sobre las certificaciones de Oracle la puede encontrar actualizada en el siguiente enlace:

https://www.oracle.com/es/corporate/features/oracle-certification.html

Los pasos necesarios para la obtención de la certificación son los siguientes:

1. Explore y seleccione la certificación adecuada.
2. Prepárese para el examen.
3. Regístrese en su examen de certificación.

Explore y seleccione la certificación adecuada

La experiencia técnica es provechosa y recomendada a la hora de afrontar un examen de certificación. Una combinación de experiencia de uso del producto, la propia experiencia técnica y la Certificación, asegurará más amplias oportunidades de carrera para usted.

Escoja su camino preferido, y repase las exigencias deseadas para afrontar la Certificación que le gustaría obtener. Las exigencias de Certificación varían para cada Certificación. Acceda a la información completa sobre la Certificación seleccionada desde la propia página de Certificaciones de Oracle, donde se informa de todos los requisitos y el temario necesario para afrontar el examen de Certificación.

Puede encontrar todas las certificaciones que posee Oracle en el siguiente enlace:

https://education.oracle.com/oracle-certification-paths-all?intcmp=WWOUOCOMCERTFEATURESTORY

Prepararse para el examen

Existe un buen número de exámenes de muestra para la preparación de las distintas Certificaciones de Oracle que se pueden obtener on-line, introduciendo el código de examen en un buscador de Internet.

También, Oracle pone a disposición de los candidatos a obtener una Certificación, un conjunto de recursos para la preparación a sus certificaciones. Pueden acceder a estos recursos en el siguiente enlace de Internet:

https://education.oracle.com/exam-preparation-packages

Registrarse y completar el examen

Visite la página web de registro de exámenes para ver las instrucciones sobre la petición de una cita en un Centro examinador de Oracle (Oracle Testing Center), o a través de un Centro examinador autorizado Pearson VUE (Pearson VUE Authorized Test Center). También es posible realizar el examen de certificación online. Toda la información la pueden encontrar en el enlace siguiente:

https://education.oracle.com/registration?intcmp=WWOUOCOMCERTFEATURESTORY

Completar los requerimientos de tu certificación

La dirección que usted facilite a Pearson VUE, en el momento del registro del examen de Certificación, será usada por Oracle para enviarle el Kit del Programa de Éxito de la Certificación de Oracle. Asegúrese de mantener al día los datos del domicilio y correo electrónico. Si necesita cambiar cualquier información de contacto, puede visitar la página https://home.pearsonvue.com/oracle.

Además, confirme que usted completó todos los requerimientos de la Certificación correspondientes al número de identificación del Test. Puede confirmar su histórico de requerimientos de certificación de manera on-line, en el enlace https://catalog-education.oracle.com/pls/apex/f?p=1010:26:104087966654139.

Preguntas tipo examen de certificación sql

En esta sección se ha recopilado de diversas fuentes públicas para la preparación de los exámenes de Certificación de Oracle, una serie de preguntas que le pueden orientar y servir de práctica a la hora de afrontar un examen de Certificación en el lenguaje SQL.

En concreto, actualmente existen los siguientes códigos de examen de Certificación en PL/SQL:

- 1Z0-149: Oracle Database PL/SQL Developer Certified Professional

Es importante reseñar que todos los exámenes se realizan en inglés, por lo que debe de estar familiarizado con la terminología técnica en dicho idioma.

Las preguntas aquí recogidas en algunos casos se han traducido al castellano para facilitar su comprensión, pero no debe confundir al candidato que se presente al examen, dado que no las encontrará así.

CUESTIÓN 1

¿Cuántas veces se ejecutará la sentencia de inserción dentro de la ejecución del FOR LOOP del siguiente bloque de PL/SQL?

```
DECLARE
      v_lower     NUMBER := 2;
      v_upper     NUMBER := 100;
      v_count     NUMBER := 1;
BEGIN
      FOR i IN v_lower..v_lower LOOP
            INSERT INTO  test(results)
            VALUES(v_count);
            v_count := v_count + 1;
      END LOOP;
END;
```

A. 0
B. 1
C. 2
D. 98
E. 100

CUESTIÓN 2

Evalúa el siguiente código PL/SQL y responde a la pregunta de ¿Cuál será el valor de V_RESULT si los 3 registros son borrados?

```
DECLARE
      V_result    BOOLEAN;
BEGIN
      DELETE FROM sale
      WHERE salesperson_id IN (25,35,45);
      V_result := SQL%ISOPEN;
      COMMIT;
END;
```

A. 0
B. 3
C. TRUE
D. NULL
E. FALSE

CUESTIÓN 3

Dado el siguiente procedimiento, si V_BONUS = TRUE y V_RAISE = NULL. ¿Qué valor se asigna a V_ISSUE_CHECK?

```
PROCEDURE dept_salary ( v_bonus           IN BOOLEAN,
                        v_raise           IN BOOLEAN,
                        v_issue_check     IN OUT BOOLEAN)
IS
BEGIN
      V_issue_check := v_bonus OR v_raise;
END;
```

A. TRUE
B. FALSE
C. NULL
D. Ninguna de las anteriores

CUESTIÓN 4

Los procedimientos y funciones son muy similares. ¿Qué razón nos hace elegir crear una función en vez de un procedimiento, en caso de que podamos hacer ambos?

A. Una función devuelve 1 valor.
B. Una función se puede usar en una sentencia SQL.
C. Una función sólo se puede ejecutar desde un procedimiento creado previamente.
D. Una función garantiza que sólo se consulta información, mientras que un procedimiento no garantiza esto.

CUESTIÓN 5

Examina la siguiente función PL/SQL.

```
CREATE OR REPLACE FUNCTION get_budget(v_studio_id IN NUMBER)
RETURN number IS
      v_yearly_budget NUMBER;
BEGIN
      SELECT yearly_budget
      INTO v_yearly_budget
      FROM studio
      WHERE id = v_studio_id;

      RETURN v_yearly_budget;
END;
```

Este procedimiento es propiedad del usuario PROD. El usuario JSMITH debe ejecutar el mismo. ¿Qué clase de privilegios debe otorgar PROD a JSMITH?

A. GRANT EXECUTE ON get_budget TO jsmith;
B. GRANT EXECUTE, SELECT ON studio TO jsmith;
C. GRANT EXECUTE, SELECT ON get_budget TO jsmith;
D. GRANT SELECT ON Studio TO jsmith;
   GRANT EXECUTE ON get_budget TO jsmith;

CUESTIÓN 6

Examina la siguiente función y determina que conjunto de sentencias invocarán la función de forma correcta dentro de un intérprete como SQL*PLUS.

```
CREATE OR REPLACE FUNCTION get_budget(v_studio_id IN NUMBER)
RETURN number IS
      v_yearly_budget NUMBER;
BEGIN
      SELECT yearly_budget
      INTO v_yearly_budget
      FROM studio
      WHERE id = v_studio_id;

      RETURN v_yearly_budget;
END;
```

A. VARIABLE g_yearly_budget NUMBER;
   : g_yearly_budget := GET_BUDGET(11);
B. VARIABLE g_yearly_budget NUMBER;
   EXECUTE g_yearly_budget := GET_BUDGET(11);
C. VARIABLE g_yearly_budget NUMBER;
   EXECUTE :g_yearly_budget := GET_BUDGET(11);
D. VARIABLE :g_yearly_budget NUMBER;
   EXECUTE :g_yearly_budget := GET_BUDGET(11);

CUESTIÓN 7

Como consecuencia de una alteración de una tabla concreta, se desea saber que procedimientos y funciones se han podido ver afectados por dicha alteración. ¿Qué tabla del sistema se ha de consultar para saberlo?

A. USER_STATUS
B. USER_SOURCE
C. USER_OBJECTS
D. USER_CONSTRUCTS

CUESTIÓN 8

Examine el siguiente código correspondiente a un paquete y responda a la pregunta siguiente: ¿Qué frase es verdadera respecto al valor de CURRENT_AVG_COST_TICKET?

```
CREATE OR REPLACE PACKAGE BODY theater_pck IS
 current_avg_cost_per_ticket   NUMBER := 0;

PROCEDURE find_cpt
 (v_movie_id IN NUMBER, v_cost_per_ticket IN OUT NUMBER)
IS
BEGIN
  IF v_cost_per_ticket > current_avg_cost_per_ticket THEN
      SELECT cost_per_ticket
      INTO v_cost_per_ticket
      FROM gross_receipt
      WHERE movie_id = v_movie_id;
  END IF;
END find_cpt;

PROCEDURE find_seats_sold
(v_movie_id IN NUMBER DEFAULT 34, v_theater_id IN NUMBER) IS
  v_seats_sold gross_receipt.seats_sold%TYPE;
  v_budget       studio.yearly_budget%TYPE;
BEGIN
      SELECT seats_sold
      INTO v_seats_sold
      FROM gross_receipt
      WHERE movie_id = v_movie_id
      AND theater_id = v_theater_id;
END find_seats_sold;

BEGIN
      current_avg_cost_per_ticket := 8.50;
END theater_pck;
```

A. Es 0 hasta que sea referenciado explícitamente.
B. Se le asigna 8.50 cada vez que el paquete es referenciado.
C. Se le asigna 8.50 únicamente cuando es referenciado explícitamente.
D. Se le asigna 8.50 cuando el paquete es invocado por primera vez dentro de una sesión.

CUESTIÓN 9

Examine el siguiente código correspondiente a un paquete y responda a la pregunta siguiente: ¿Cuál sería la correcta invocación del procedimiento FIND_SEATS_SOLD de dicho paquete si lo queremos hacer directamente desde el prompt de SQL*PLUS?.

```
CREATE OR REPLACE PACKAGE BODY theater_pck IS
      v_total_budget  NUMBER;

PROCEDURE find_seats_sold
(v_movie_id IN NUMBER DEFAULT 34, v_theater_id IN NUMBER);
END theater_pck;


CREATE OR REPLACE PACKAGE BODY theater_pck IS
      current_avg_cost_per_ticket   NUMBER;

PROCEDURE find_seats_sold
(v_movie_id IN NUMBER DEFAULT 34, v_theater_id IN NUMBER) IS
      v_seats_sold gross_receipt.seats_sold%TYPE;
      v_budget       studio.yearly_budget%type;
BEGIN
      SELECT seats_sold
      INTO v_seats_sold
      FROM gross_receipt
      WHERE movie_id = v_movie_id
      AND theater_id = v_theater_id;
END find_seats_sold;

FUNCTION get_budget(v_studio_id IN NUMBER)
RETURN number IS
      v_yearly_budget NUMBER;
BEGIN
      SELECT yearly_budget
      INTO v_yearly_budget
      FROM studio
      WHERE id = v_studio_id;
      RETURN v_yearly_budget;
END get_budget;

END theater_pck;
```

A. EXECUTE find_seats_sold(500,11);
B. Theater_pck.find_seats_sold(500,11);
C. EXECUTE theater_pck.find_seats_sold(500,11);
D. SELECT find_seats_sold(movie_id, theatre_id) FROM gross_receipt;

CUESTIÓN 10

Examina el siguiente código y selecciona 1 respuesta como única correcta.

```
CREATE OR REPLACE PACKAGE PROD_PACK IS
      G_TAX_RATE  NUMBER  := .08;
END PROD_PACK;
```

A. Esta especificación de cabecera de paquete, puede existir sin un cuerpo de paquete.
B. Este cuerpo de paquete puede existir sin una especificación de cabecera de paquete.
C. Este cuerpo de paquete no puede existir sin una especificación de cabecera de paquete.
D. Esta especificación de cabecera de paquete no puede existir sin un cuerpo de paquete.

CUESTIÓN 11

Examina el siguiente código y di que frase es correcta sobre los procedimientos definidos en esta especificación de cabecera de paquete:

```
CREATE OR REPLACE PACKAGE theater_package IS
PROCEDURE find_cpt (v_movie_id           IN NUMBER,
                    v_cost_per_ticket    IN OUT NUMBER);
PROCEDURE update_theater (v_name IN VARCHAR2);
PROCEDURE find_seats_sold (v_movie_id    IN NUMBER DEFAULT 34,
                           v_theater_id  IN NUMBER);
PROCEDURE add_theater;
END theater_package;
```

A. Todos los procedimientos son construcciones públicas.
B. Todos los procedimientos son construcciones privadas.
C. Cada construcción de procedimiento tiene omitido el código, por tanto es ilegal.
D. Cada construcción de procedimiento contiene una lista de argumentos, por tanto es ilegal.

CUESTIÓN 12

Examina este trigger:

```
CREATE OR REPLACE TRIGGER audit_gross_receipt
AFTER DELETE OR UPDATE OF seats_sold, cost_per_ticket ON
gross_receipt
BEGIN

{additional code}

END;
```

¿Cuántas veces se ejecutará el cuerpo del trigger por cada invocación?

A. Una vez.
B. Dos veces.
C. 1 vez por cada fila borrada o actualizada.
D. 1 vez por cada fila borrada o se actualice SEATS_SOLD o COST_PER_TICKET.

CUESTIÓN 13

Examine este trigger:

```
CREATE OR REPLACE TRIGGER update_studio
BEFORE UPDATE OF yearly_budget ON STUDIO
FOR EACH ROW
```

¿Qué evento invocará al trigger?

A. Una actualización de la tabla STUDIO.
B. Cualquier modificación de la tabla STUDIO.
C. Una actualización de la columna YEARLY_BUDGET.
D. Una actualización de cualquier columna que no sea YEARLY_BUDGET.

CUESTIÓN 14

Given this PL/SQL block:

```
BEGIN
       INSERT INTO employee(salary, last_name, first_name)
       VALUES(35000, 'Wagner', 'Madeline');
       SAVEPOINT save_a;
       INSERT INTO employee(salary, last_name, first_name)
       VALUES(40000, 'Southall', 'David');
       SAVEPOINT save_b;
       DELETE FROM employee
       WHERE dept_no = 10;
       SAVEPOINT save_c;
       INSERT INTO employee(salary, last_name, first_name)
       VALUES(25000, 'Brown', 'Bert');
       ROLLBACK TO SAVEPOINT save_c;
       INSERT INTO employee(salary, last_name, first_name)
       VALUE(32000, 'Dean', 'Mike');
       ROLLBACK TO SAVEPOINT save_b;
       COMMIT;
END;
```

Which two changes to the database will be made permanent? (Choose two.)

A. DELETE FROEM employee WHERE dept_no = 10;
B. INSERT INTO employee(salary, last_name, first_name)
   VALUES(32000, 'Dean', 'Mike');
C. INSERT INTO employee(salary, last_name, first_name)
   VALUES(25000, 'Brown', 'Bert');
D. INSERT INTO employee(salary, last_name, first_name)
   VALUES(40000, 'Southall', 'David');
E. INSERT INTO employee(salary, last_name, first_name)
   VALUES(35000, 'Wagner', 'Madeline');

CUESTIÓN 15

Arrange the events that occur when an explicit cursor is opened and one row is fetched in the appropriate order.

When an explicit cursor is opened and one row is fetched the following events occur:

1. The PL/SQL variables are populated.
2. The active set is identified.
3. The pointer is advanced.
4. A query is executed.
5. The current row data is read.
6. The pointer is positioned before the first row.

A. 1,2,3,4,5,6
B. 5,3,2,4,6,1
C. 2,4,5,3,1,6
D. 4,2,6,3,5,1

CUESTIÓN 16

Which does NOT happen when rows are found using a FETCH statement?

A. The cursor remains open after each fetch.
B. The active set is identified satisfying the search criteria.
C. Output PL/SQL variables are populated with the current row data.
D. The pointer identifying the next row in the active set is advanced.

CUESTIÓN 17

Examine this database trigger:

```
CREATE OR REPLACE TRIGGER update_studio
BEFORE UPDATE OF yearly_budget ON STUDIO
FOR EACH ROW
DECLARE
      v_max_budget NUMBER;
BEGIN
      SELECT max(yearly_budget)
      INTO v_max_budget
      FROM studio;
      IF :new.yearly_budget > v_max_budget THEN
            :new.yearly_budget := v_max_budget;
      END IF;
END;
```

After creating this database trigger successfully, you test it by updating the YEARLY_BUDGET column to a value greater than the maximum value already in the STUDIO table. What result can you expect?

The YEARLY_BUDGET column will be set to the maximun value.

A. The STUDIO table is mutating and therefore, an error is returned.
B. The STUDIO table is mutating and therefore, the trigger execution is ignored.
C. Referencing the NEW qualifier in a BEFORE trigger is illegal and therefore, an error is returned.

CUESTIÓN 18

The script file containing the CHECK_SAL server procedure is lost. This procedure must be modified immediately and recreated. Which data dictionary view can you query to retrieve the source code of this procedure?

A. ALL_SOURCE
B. ALL_PROCEDURES
C. PROCEDURE_SOURCE
D. SERVER_PROCEDURES

CUESTIÓN 19

¿Que 2 sentencias referidas a la sobrecarga de paquetes es verdadera?

A. Los subprogramas deben de ser locales.
B. Los subprogramas pueden ser locales o remotos.
C. Está permitido exceder el máximo número de subprogramas.
D. Dos subprogramas con el mismo nombre deben diferir sólo en el tipo que se devuelve.
E. Dos subprogramas con el mismo nombre y número de parámetros formales deben tener al menos un parámetro definido con el tipo de datos diferente.

CUESTIÓN 20

Examine this database trigger:

```
CREATE OR REPLACE TRIGGER audit_gross_receipt
AFTER DELETE OR UPDATE OF seats_sold, cost_per_ticket ON
gross_receipt
BEGIN
{additional code}
END;
```

If a user issues a DELETE statement against the GROSS_RECEIPT table, the procedure, AUDIT_DELETE_GR of the THEATER_PCK package, must be executed once for the entire DELETE statement. Which code, when added, will perform this successfully?

A. IF DELETING THEN theatre_pck.audit_delete_gr;
   END IF;
B. IF CHECK_DELETE THEN theatre_pck.audit_delete_gr;
   END IF;
C. IF DBMS_SQL('DELETE') THEN theatre_pck.audit_delete_gr;
   END IF;
D. IF DMBS_CHECK_MANIPULATION('DELETE') THEN theatre_pck.audit_delete_gr;
   END IF;

CUESTIÓN 21

Which data dictionary table can you query to view all the dependencies between the objects that you own?

A. USER_OBJECTS
B. USER_RELATIONS
C. USER_DEPENDENCIES
D. USER_RELATIONSHIPS

CUESTIÓN 22

Examine this function:

```
CREATE OR REPLACE FUNCTION set_budget (v_studio_id IN NUMBER,
                                       v_new_budget IN NUMBER)
RETURN number IS
BEGIN
   UPDATE studio
   SET yearly_budget = v_new_budget
   WHERE id = v_studio_id;
   COMMIT;
      RETURN SQL%ROWCOUNT;
END;
```

This function is executed from within a procedure called CALCULATE_BUDGET. The database administrator has just informed you that a new column has been added to the STUDIO table. What affect will this have?

A. Only SET_BUDGET will be marked invalid.
B. Only CALCULATE_BUDGET will be marked invalid.
C. Both SET_BUDGET and CALCULATE_BUDGET will be marked invalid.
D. SET_BUDGET, CALCULATE_BUDGET, and STUDIO will be marked invalid.

CUESTIÓN 23

Which character function would you use to return a specified portion of a character string?

A. CONCAT
B. SUBSTR
C. LENGTH
D. INITCAP

CUESTIÓN 24

You attempt to create the ALPHA_3000 table with this statement:

```
1.  CREATE TABLE alpha_3000
2.   (3000_id     NUMBER(9)
3.                CONSTRAINT alpha_3000_id_pk PRIMARY KEY,
4.    name        VARCHAR2(25),
5.    title       VARCHAR2(25),
6.    idname      VARCHAR2(25)
7.                CONSTRAINT alpha_3000_idname_nn NOT NULL);
```

Which line in the statement causes a syntax error?

A. 1
B. 2
C. 3
D. 7

CUESTIÓN 25

Evaluate this PL/SQL block:

```
DECLARE
     v_result          BOOLEAN;
BEGIN
     DELETE FROM sale WHERE salesperson_id IN (25, 35, 45);
     v_result := SQL%ISOPEN;
     COMMIT;
END;
```

What will be the value of V_RESULT if three rows are deleted?

A. 0
B. 3
C. TRUE
D. NULL
E. FALSE

CUESTIÓN 26

Which SELECT statement is an equijoin query between two tables?

A. SELECT region.region_name, employee.salary
   FROM region, employee
   WHERE region.id = employee.region_no;
B. SELECT region.region_name, employee.salary
   FROM region, employee
   WHERE region.id = employee.region_no(+);
C. SELECT region.region_name, employee.salary
   FROM region, employee
   WHERE employee.salary BETWEEN region.avq_salary AND region.max_salary;
D. SELECT region.region_name, employeeinfo.last_name
   FROM employee region, employee.employeeinfo
   WHERE employeeinfo.id >= region.manager_id;

CUESTIÓN 27

The CUSTOMER table has been created. You attempt to create the SALE table with this command:

```
1.   CREATE TABLE sale
2.   (purchase_no     NUMBER(9),
3.    customer_no     NUMBER(9)
4.    CONSTRAINT sale_customer_id_fk REFERENCES
5.              customer (id),
6.    CONSTRAINT sale_purchase_no_pk PRIMARY KEY (purchase_no),
7.    CONSTRAINT sale_customer_no_nn NOT NULL (customer_no));
```

Which line in the statement will cause an error?

A. 2
B. 3
C. 4
D. 6
E. 7

CUESTIÓN 28

The TRANSACTION table has six columns. Since you often query the table with a join to the SALE table, you created an index on five of the columns in the TRANSACTION table. Which result will occur?

A. Inserts to the table will be slower.
B. The speed of deletes will be increased.
C. The size of the TRANSACTION table will be increased.
D. All queries on the TRANSACTION table will be faster if it does contain a large number of NULL values.

CUESTIÓN 29

You alter the database with this command:

RENAME streets TO city;

Which task is accomplished?

A. The STREETS user is renamed CITY.
B. The STREETS table is renamed CITY.
C. The STREETS column is renamed CITY.
D. The STREETS constraint is renamed CITY.

CUESTIÓN 30

You query the database with this command:

```
SELECT name, salary, dept_id
FROM employee
WHERE salary >
        (SELECT AVG(salary)
         FROM employee
         WHERE dept_no =
              (SELECT dept_no
               FROM employee
               WHERE last_name =
                 (SELECT last_name
                  FROM employee
                  WHERE salary > 50000)));
```

Which SELECT clause is evaluated first?

A. SELECT dept_no
B. SELECT last_name
C. SELECT AVG(salary)
D. SELECT name, salary, dept_id

CUESTIÓN 31

Evaluate this PL/SQL block:

```
BEGIN
      FOR i IN 1..6 LOOP
            IF i = 1 THEN
                  COMMIT;
            ELSE
                  IF i = 3 THEN
                        ROLLBACK;
                  ELSE
                        IF i = 5 THEN
                              COMMIT;
                        ELSE
                              INSERT INTO  exam(id)
                              VALUES            (i);
                        END IF;
                  END IF;
            END IF;
      END LOOP;
      COMMIT;
END;
```

How many values will be inserted into the EXAM table?

A. 0
B. 1
C. 2
D. 3
E. 5
F. 6

CUESTIÓN 32

You query the database with this command:

```
SELECT name
FROM employee
WHERE name LIKE '_a%';
```

Which names are displayed?

A. Names starting with "a".
B. Names starting with "a" or "A".
C. Names containing "a" as second letter.
D. Names containing "a" as any letter except the first.

CUESTIÓN 33

Evaluate this SQL statement:

```
SELECT id, (2 * cost) / (2 * sale_price) + 10 price
FROM   product;
```

All of the COST and SALE_PRICE values in the PRODUCT table are greater than one. What would happen if you removed all the parentheses from the calculation?

A. The statement would generate a syntax error.
B. The statement would achieve the same results.
C. The results of the PRICE values would be lower.
D. The results of the PRICE values would be higher.

CUESTIÓN 34

Evaluate this IF statement. You issue this command:

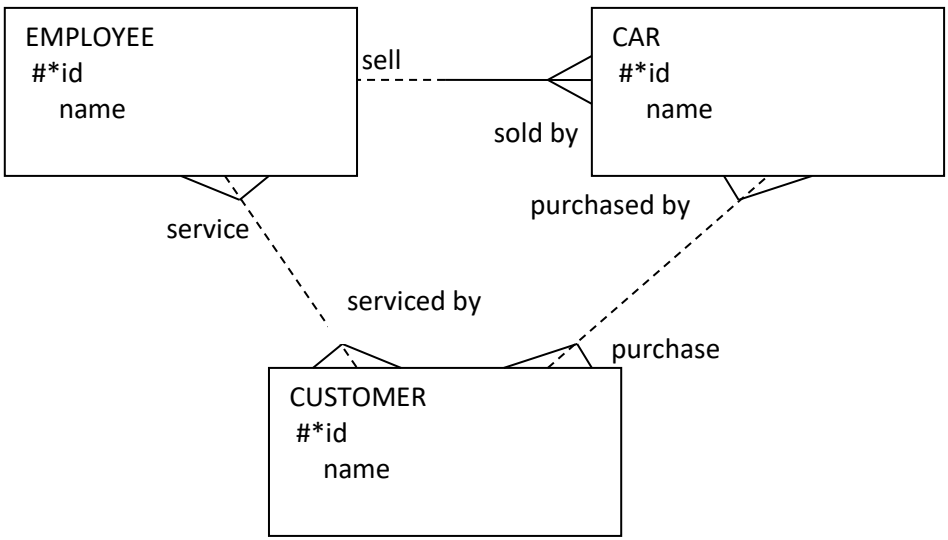
```
GRANT update
ON employee
TO ed
WITH GRANT OPTION;
```

Which task could Ed perform on the EMPLOYEE table?

A. View data.
B. Delete data.
C. Modify constraint.
D. Give update access to other users.

CUESTIÓN 35

Based on the diagram, which relationship is mandatory?



A. Employees sell cars.
B. Customers purchase cars.
C. Cars are sold by employees.
D. Employees service customers.

CUESTIÓN 36

You query the database with this command:

```
SELECT object_name
FROM all_objects
WHERE object_type = 'TABLE';
```

Which values are displayed?

A. Only the names of the tables you own.
B. Only the names of the objects you own.
C. Only the names of all the objects you can access.
D. Only the names of all the tables you can access.

CUESTIÓN 37

What happens when rows are found using a FETCH statement?

A. The cursor opens.
B. The cursor closes.
C. The current row values are loaded into variables.
D. Variables are created to hold the current row values.

CUESTIÓN 38

What is the maximum number of handlers processed before the PL/SQL block is exited when an exception occurs?

A. Only one.
B. All referenced.
C. All that apply.
D. None.

CUESTIÓN 39

Evaluate this SQL script:

```
CREATE ROLE payroll;
CREATE ROLE sales_dept;
CREATE ROLE inventory;
CREATE USER scott IDENTIFIED BY tiger;
GRANT SELECT ON employee TO payroll;
GRANT SELECT ON sale TO sales_dept;
GRANT payroll TO sales_dept;
GRANT sales_dept TO inventory;
GRANT inventory TO scott
/
```

Which tables can user SCOTT query?

A. Only SALE.
B. Only EMPLOYEE.
C. Both SALE and EMPLOYEE.
D. Neither SALE nor EMPLOYEE.

CUESTIÓN 40

You issue this command:

```
SELECT emp_id_seq.CURRVAL
FROM SYS.dual;
```

Which value(s) is displayed?

A. Values of the EMP_ID_SEQ column.
B. Current value of the EMP_ID_SEQ index.
C. Current value of the EMP_ID_SEQ cursor.
D. Current value of the EMP_ID_SEQ sequence.

CUESTIÓN 41

Which program construct must return a value?

A. Package.
B. Function.
C. Anonymous block.
D. Stored procedure.
E. Application procedure.

CUESTIÓN 42

You issue this command:

```
ALTER USER ed IDENTIFIED BY wvu88;
```

Which task has been accomplished?

A. A new user has been added.
B. The user name has been changed.
C. The user password has been changed.
D. A password has been added to the user account.

CUESTIÓN 43

Peter works as a Database Administrator. He create a table named ACCOUNTS that contains the accounts information of the company. Peter wants to validate data automatically, before it is inserted into the ACCOUNTS table. Which of the following will Peter use to accomplish this task?

A. Stored procedure.
B. Anonymous block
C. SELECT statement
D. Trigger

CUESTIÓN 44

You work as an Application Developer in a company. The company uses an Oracle database. The database contains a table names EMPLOYEES. The EMPLOYEES table contains a column of the LONG datatype. You want to change the datatype of the column from LOG to BLOB. What will you do to accomplish this?

A. Use the DBMS_LOG.MIGRATE stored procedure.
B. Use the ALTER DATABASE statement.
C. Use the ALTER TABLE statement.
D. You cannot accomplish the task.

CUESTIÓN 45

David is an employee in Technet Inc. The company uses an Oracle database. David has been granted a role names SALES. David wants to DROP the role. He executes the following statement to accomplish this:

```
DROP ROLE Sales;
```

When he executes the statement, it returns an error. What is the most likely cause of the issue?

Each correct answer represents a complete solution. Choose two.

A. David is not granted the role with the GRANT OPTION.
B. David is not granted the role with de ADMIN OPTION.
C. Only the database administrator can drop the role.
D. David does not have the DROP ANY ROLE system privilege.

CUESTIÓN 46

You work as an Application Developer for Technet Inc. The company uses an Oracle database. The database contains a table names EMPLOYEES. You have defined a database trigger named RAISE_SALARY on the EMPLOYEES table. You want to remove the trigger from the database. Which of the following SQL statements will you use to accomplish this?

A. ALTER TRIGGER Raise_Salary REMOVE;
B. DELETE TRIGGER Raise_Salary;
C. DROP TRIGGER Raise_Salary;
D. REMOVE TRIGGER Raise_Salary;

CUESTIÓN 47

Under which of the following conditions is the use of an explicit cursor necessary?

A. When any SQL data manipulation language (DML) statement is used.
B. When a query returns only one row.
C. When a query does not return any row.
D. When a query returns more than one row.

CUESTIÓN 48

Which of the following statements about a package are true?

Each correct answer represents a complete solution. Choose two.

A. The specification of a package declares the package constructs, whereas the body of the package defines them.
B. A package has three parts.
C. A package itself cannot be called or nested.
D. A package is loaded into the memory every time a package construct is called.

CUESTIÓN 49

Which of the following functions is not available in procedural statements?

A. MOD
B. TRUNC
C. TO_DATE
D. TO_CHAR
E. LOWER
F. DECODE

CUESTIÓN 50

Which of the following is a cursor attribute that yields the number of rows processed by the last DML statement?

A. SQL%ROWRECKON
B. SQL%ROWTALLY
C. SQL%ROWCOUNT
D. SQL%ROWADD

CUESTIÓN 51

Identify whether the given statement is true or false?

"Only the IN parameters can be assigned a default value. The OUT and IN OUT parameters cannot be assigned a default value."

A. False
B. True

CUESTIÓN 52

Which of the following SQL*Plus commands is used to invoke a procedure from iSQL*Plus?

A. RUN
B. EXECUTE
C. CALLL
D. REVOKE
E. START
F. STARTUP

CUESTIÓN 53

Identify whether the given statement is true or false? "A stored procedure cannot be executed as a stand-alone statement."

A. True
B. False

CUESTIÓN 54

Which of the following DBMS_SQL Subprograms combine a given value to a given collection?

A. ADD_ARRAY Procedures
B. UNITE_ARRAY Procedures
C. COMBINE_ARRAY Procedures
D. BIND_ARRAY Procedures

CUESTIÓN 55

Which of the following methods removes one element form the end of a collection?

A. DELETE(m,n)
B. TRIM(n)
C. DELETE(n)
D. DELETE
E. TRIM

Antolín·Muñoz·Chaparro¶
Jefe·de·Proyecto·Sistemas·Informáticos·¶
División·III·de·Aplicaciones·de·Costes·de·Personal·y·Pensiones·Públicas¶
Oficina·de·Informática·Presupuestaria·(DIR3:·EA0027952)¶
Intervención·General·de·la·Administración·del·Estado¶