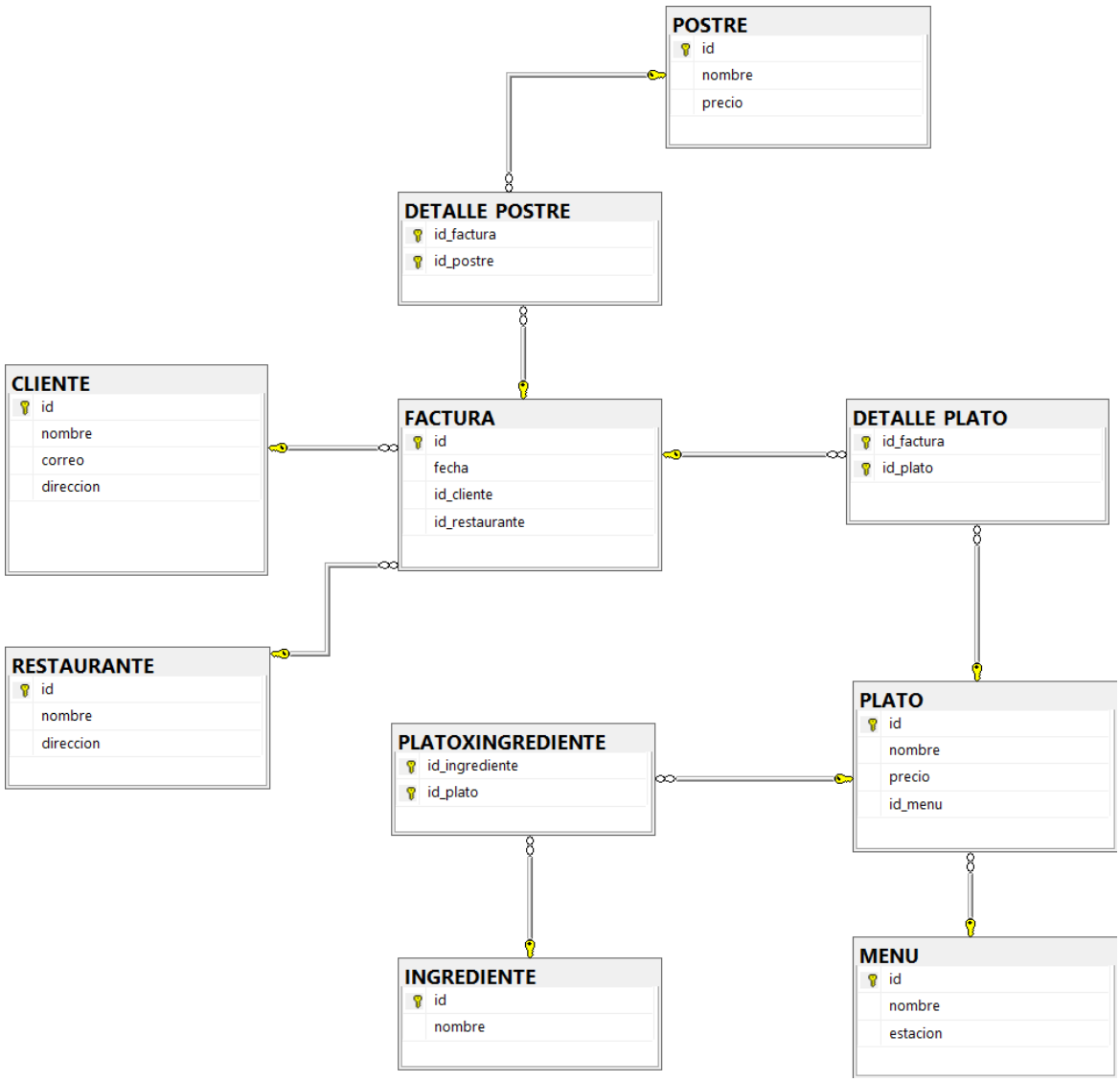
Universidad Centroamericana José Simeón Cañas
Departamento de electrónica e informática
Administración de bases de datos, 2024
Programación PL/SQL.

Diagrama relacional ejercicio 1, 2, 3 y 4.



Ejercicio 1.

Basándose en la siguiente consulta:

```
SELECT F.ID, F.fecha, F.id_cliente, F.id_restaurante, SUM(P.precio) "Subtotal plato" FROM FACTURA F
    LEFT JOIN DETALLE_PLATO DP ON DP.ID_FACTURA = F.ID
    LEFT JOIN PLATO P ON P.ID = DP.ID_PLATO
GROUP BY F.ID, F.fecha, F.id_cliente, F.id_restaurante;
```

Crear una función que reciba como parámetro el id de una factura la función retornará el subtotal calculado a partir de los platillos consumidos en el servicio. Luego, Basándose en la siguiente consulta:

```
SELECT F.ID, F.fecha, F.id_cliente, F.id_restaurante, SUM(P.precio) "Subtotal postre" FROM FACTURA F
    LEFT JOIN DETALLE_POSTRE DP ON DP.ID_FACTURA = F.ID
    LEFT JOIN POSTRE P ON P.ID = DP.ID_POSTRE
GROUP BY F.ID, F.fecha, F.id_cliente, F.id_restaurante;
```

Crear una función que reciba como parámetro el id de una factura la función retornará el subtotal calculado a partir de los postres consumidos en el servicio.

Ejercicio 2.

Crear una función que reciba como parámetros valores que representan dos fechas, el tipo de dato debe ser DATE. La función retornará la lista de facturas registradas en la base de datos en el rango de fechas definidos como parámetros. El resultado debe incluir el subtotal de platos, el subtotal de postres y el total (suma del subtotal de platos + subtotal de postres). Para calcular los subtotales, la función debe hacer uso de las dos funciones creadas en el ejercicio 1.

Resultado esperado: si se ejecuta la función entre el 1 de enero de 2022 y el 30 de enero de 2022, el resultado debería ser:

	id	fecha	id_cliente	id_restaurante	subtotal_plato	subtotal_postre	total
1	6	2022-01-06	3	5	50.31	14.60	64.91
2	20	2022-01-07	15	1	24.22	13.75	37.97
3	25	2022-01-03	9	4	63.27	5.56	68.83
4	28	2022-01-11	10	3	97.22	23.10	120.32
5	29	2022-01-27	2	2	43.45	3.15	46.60

Ejercicio 3.

Crear un procedimiento almacenado que reciba como parámetros el id de un restaurante (INT), y dos fechas (VARCHAR2). El procedimiento deberá hacer uso de la función creada en el ejercicio 2 para poder obtener las ganancias de un restaurante específico en un rango de fechas, luego, se deberá recorrer el resultado de la consulta con un cursor para imprimir su contenido en consola. El procedimiento deberá hacer uso correcto de cursores y bloques TRY/CATCH para realizar el procesamiento de datos.

Sugerencia:
Declarar el curso de la siguiente forma:

```
DECLARE
CURSOR cursor_eje4 IS <CONSULTA SQL>;
num_rows NUMBER;
v_col1 <TIPO>    --Sustituir tipo por el tipo de dato correspondiente a la
columna
v_col2 <TIPO>
...
```

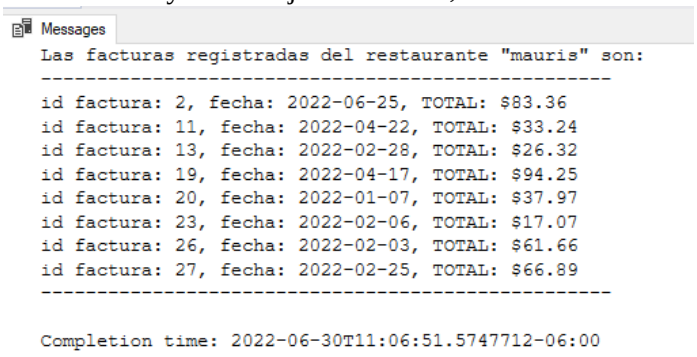
Para obtener la cantidad de filas de un cursor y poder recorrerlo es posible utilizar la siguiente opción:

```
…
BEGIN
OPEN CURSOR_EJE4;
SELECT COUNT(*) INTO num_rows FROM (<CONSULTA SQL>);

LOOP
     FETCH cursor_eje4 INTO v_col1, v_col2…;
     ...
     EXIT WHEN cursor_eje4%NOTFOUND;
END LOOP;

CLOSE CURSOR_EJE4;
END;
```

Resultado esperado: si el procedimiento almacenado se ejecuta para el restaurante con id 1, y fechas entre el 1 de enero de 2022 y el 30 de junio de 2022, el resultado debería ser:



Ejercicio 4.
Crear un TRIGGER que verifique si la estación de un plato que se desea asignar a una factura corresponde con la fecha de la factura. Cada plato tiene asignado un menú, este a su vez corresponde con una estación (es posible verificar esto en la tabla MENU).

Estación	Rango de fechas
“primavera”	Del 21 de marzo al 21 de junio
“verano”	Del 22 de junio al 23 de septiembre
“otoño”	Del 24 de septiembre al 21 de diciembre
“invierno”	Del 22 de diciembre al 20 de marzo

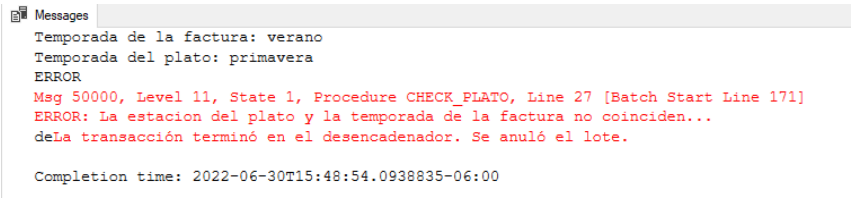
Ejemplo de ejecución:
El procedimiento es el siguiente: la primera etapa es insertar una factura, por ejemplo:

```
INSERT INTO FACTURA VALUES (31, TO_DATE('05/07/2022', 'DD/MM/YYYY'),
3, 2);
```

La fecha de la factura corresponde al 5 de julio de 2022, por lo que en la tabla PLATO_DETALLE solo se podrían insertar platos correspondientes al menú de "verano". En caso contrario el trigger provocará un error. Si se intenta insertar la siguiente información en la tabla PLATO_DETALLE:

```
INSERT INTO DETALLE_PLATO VALUES (31, 1);
```

La consulta finalizará con un erro debido a que el plato con id 1 corresponde al menú de "primavera"



Si se intenta insertar la siguiente información en la tabla PLATO_DETALLE:

```
INSERT INTO DETALLE_PLATO VALUES (31, 12);
```

La ejecución finalizará exitosamente porque el plato con id 12 pertenece al menú de "verano" al igual que la fecha de la factura.

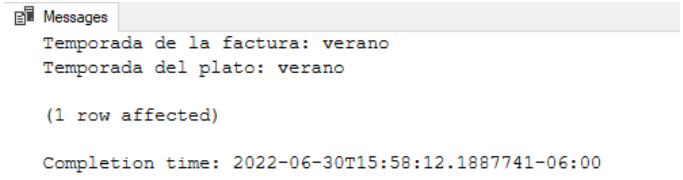
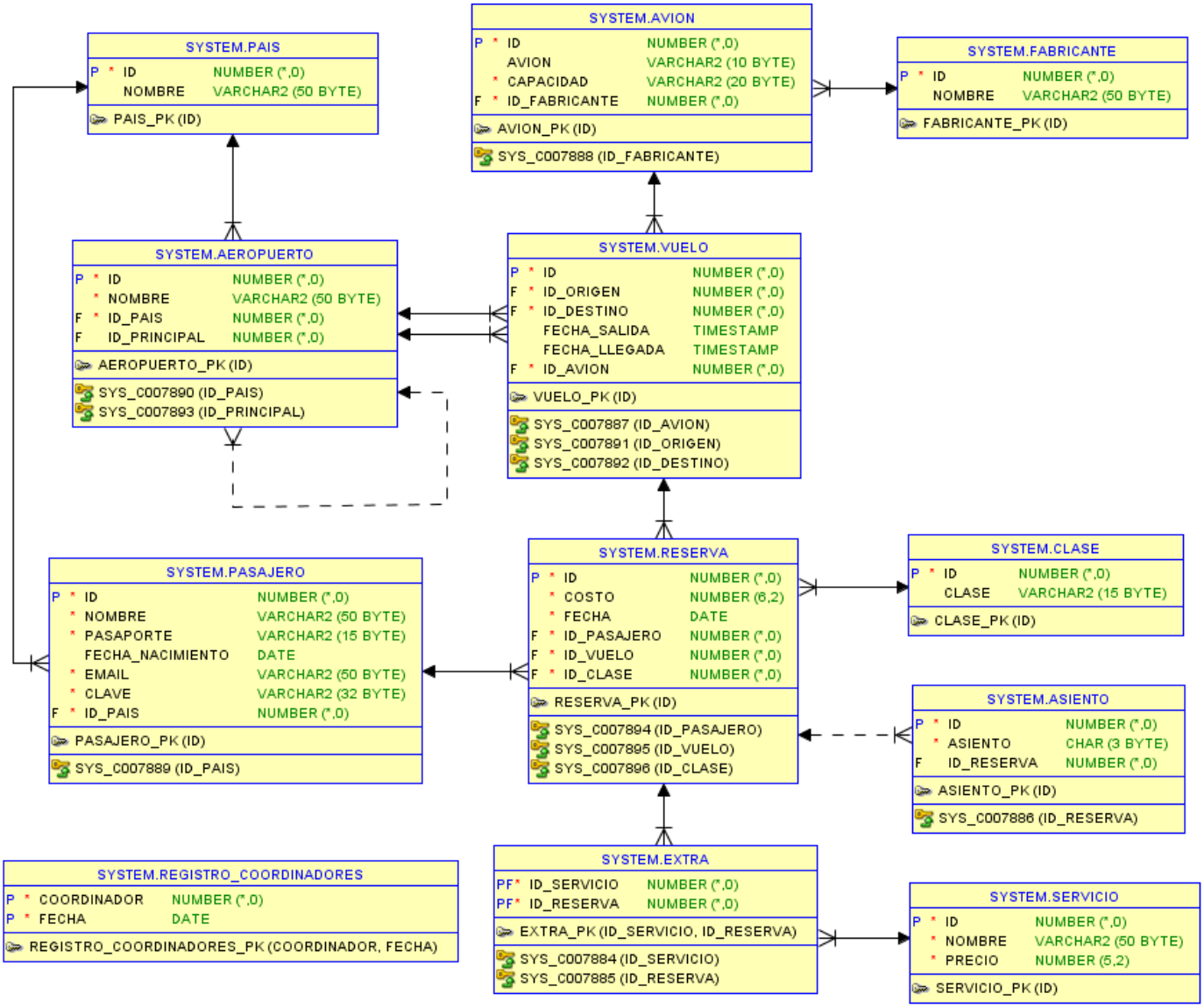


Diagrama relacional ejercicio 5, 6, 7 y 8.



Contextualización.

Una compañía que se encarga de gestionar varios aeropuertos alrededor del mundo desea establecer una nueva estrategia de coordinación entre los distintos aeropuertos. Para ello, cada trimestre se seleccionará de forma aleatoria 3 aeropuertos que funcionarán como "aeropuertos de coordinación". Estos aeropuertos se asignarán también de forma aleatoria en el resto de los aeropuertos, de tal forma, que cada aeropuerto tenga asignado un aeropuerto de coordinación. Debido a este requerimiento, en la tabla AEROPUERTO, se ha creado una columna extra llamada "id_principal" en donde se planea realizar la asignación de los aeropuertos de coordinación. Para poder aplicar esta funcionalidad se requiere realizar las siguientes tareas:

Ejercicio 5.

Crear una función que retorne un arreglo de enteros de 3 posiciones. La función se encargará de seleccionar los id de 3 aeropuertos al azar y los almacenará en el arreglo que se retornará. Se debe considerar que los rangos de selección no deben ser "quemados", sino que se debe verificar en la base de datos cuales es el primer y último id en la tabla AEROPUERTO. NOTA: se puede asumir que todos los id entre el primero y el ultimo si existen en la base de datos. Nota. En el archivo "P1. Script solucion.sql" se encuentra disponible un ejemplo sobre como crear, llenar, recorrer y retornar un arreglo desde una función y otro ejemplo sobre como generar números aleatorios dentro de un rango específico en Oracle.

Ejercicio 6.

Ahora que se dispone de una función que permite generar los aeropuertos de coordinación, es necesario crear un procedimiento almacenado que asigne de forma aleatoria los 3 aeropuertos de coordinación disponibles, por lo que este procedimiento utilizará la información que retorna la función de la tarea 1 para realizar su trabajo. Es necesario que se valide que cuando llegue el turno de un aeropuerto que ha sido seleccionado como coordinador, el valor en el campo "id_principal" sea NULL. **Nota**. se recomienda incluir al principio del procedimiento almacenado una consulta UPDATE que actualice con el valor de NULL la columna "id_principal" de todos los aeropuertos, en caso de que se desea ejecutar más de una vez, esto para mantener la consistencia en el resultado.
Luego del trabajo realizado por el procedimiento almacenado de la tarea 2, si usted realiza una consulta a los datos de la tabla AEROPUERTO, se debería observar un resultado como el de la figura 2. Todos los aeropuertos a excepción de los seleccionados como coordinadores tienen asignado un valor en el campo "id_principal". Este resultado cambiará en cada ejecución porque la asignación ha sido realizada de forma aleatoria.

	ID	NOMBRE	ID_PAIS	ID_PRINCIPAL
1	1	El Valle	15	2
2	2	Oranjestad	11	(null)
3	3	Hamilton	8	2
4	4	Kralendijk	9	10
5	5	Willemstad	13	15
6	6	Nuuk	10	2
7	7	Basse-Terre	2	2
8	8	Cayena	9	2
9	9	George Town	15	10
10	10	King Edward Point	15	(null)
11	11	Puerto Argentino	7	2
12	12	Mountainburn Town	16	2
13	13	Road Town	13	2
14	14	Carlota Amalia	7	10
15	15	Brades	9	(null)
16	16	San Juan	4	10
17	17	The Bottom	7	2
18	18	Gustavia	4	2
19	19	Oranjestad	5	2
20	20	San Pedro	10	10

Figura 1. Consulta SELECT realizada luego de la ejecución del procedimiento almacenado de la tarea 2.

Ejercicio 7.

El tercer objetivo es crear un trigger que registre en la tabla REGISTRO_COORDINADORES los 3 aeropuertos coordinadores seleccionados, se ha planificado que el trigger se ejecute cuando el procedimiento almacenado de la tarea 2 actualice la información de los aeropuertos. Cada vez que se actualice una fila de la tabla AEROPUERTO, el trigger se activará y deberá validar si se está actualizando el valor del campo “id_principal”, si es así, entonces se debe almacenar el valor de “id_principal” y la fecha en la tabla REGISTRO_COORDINADORES.

Nota: Se debe validar que cada id de aeropuerto coordinador se ingrese en la tabla REGISTRO_COORDINADORES solo una vez para la fecha en que se registra, notar que la PK de la tabla REGISTRO_COORDINADORES es una llave primaria (PK) compuesta por los campos “coordinador” y “fecha”.

Al finalizar el trabajo del trigger, la tabla REGISTRO_COORDINADORES deberá contener un resultado parecido al que se muestra en la figura 3. Nótese como cada id se ha almacenado solo una vez. Con esta metodología de trabajo, se puede asumir que los últimos 3 registros en la tabla REGISTRO_COORDINADORES son los últimos 3 aeropuertos actualmente seleccionados como coordinadores.

Nota. Los valores que almacena la tabla solo aplican para el ejemplo ejecutado en este documento, los aeropuertos coordinadores se seleccionan de forma aleatoria por lo que se espera que en cada ejecución los valores cambien. Hay que recordar que si se desea volver a ejecutar el procedimiento almacenado se debe borrar el contenido de la tabla REGISTRO_COORDINADORES para evitar problemas de llaves PK duplicadas (porque la llave es compuesta con la fecha actual del sistema), opcionalmente, se puede cambiar la fecha del sistema (se supone que el procedimiento se ejecutará trimestralmente).

	COORDINADOR	FECHA
1	2	08/09/21
2	10	08/09/21
3	15	08/09/21

Figura 2. Almacenamiento de los id de aeropuertos coordinadores en la tabla REIGSTRO_COORDINADORES.

Ejercicio 8.

Finalmente, se debe crear una función que se encargue mostrar los resultados obtenidos luego de ejecutar el procedimiento almacenado de la tarea 2. La función recibirá como parámetro un número entero, que representa el id de un aeropuerto coordinador, luego la función retornará la lista de todos los aeropuertos coordinados por el aeropuerto en cuestión.

La siguiente figura muestra el resultado que retorna la función cuando se ingresa el id del aeropuerto 2, que, en el ejemplo seguido en este documento, se trata de un aeropuerto coordinador.

ID	AEROPUERTO	PAIS	ID_COORDINADOR	AEROPUERTO_COORDINADOR
1	El Valle	Guyana	2	Oranjestad
3	Hamilton	Panamá	2	Oranjestad
6	Nuuk	Bolivia	2	Oranjestad
7	Basse-Terre	Belice	2	Oranjestad
8	Cayena	Argentina	2	Oranjestad
11	Puerto Argentino	Nicaragua	2	Oranjestad
12	Mountainburn Town	Guyana Francesa	2	Oranjestad
13	Road Town	Colombia	2	Oranjestad
17	The Bottom	Nicaragua	2	Oranjestad
18	Gustavia	El Salvador	2	Oranjestad
19	Oranjestad	Guatemala	2	Oranjestad

Figura 3. Resultado luego de ejecutar la función de la tarea 4.