



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

PRUEBAS SELECTIVAS DE ACCESO AL
GRUPO A, SUBGRUPO A2

SECTOR ADMINISTRACIÓN ESPECIAL

**CATEGORÍA/ESCALA TÉCNICO o
TÉCNICA MEDIOS DE GESTIÓN
(SISTEMAS DE INFORMACIÓN)**

SISTEMA DE CONCURSO-OPOSICIÓN

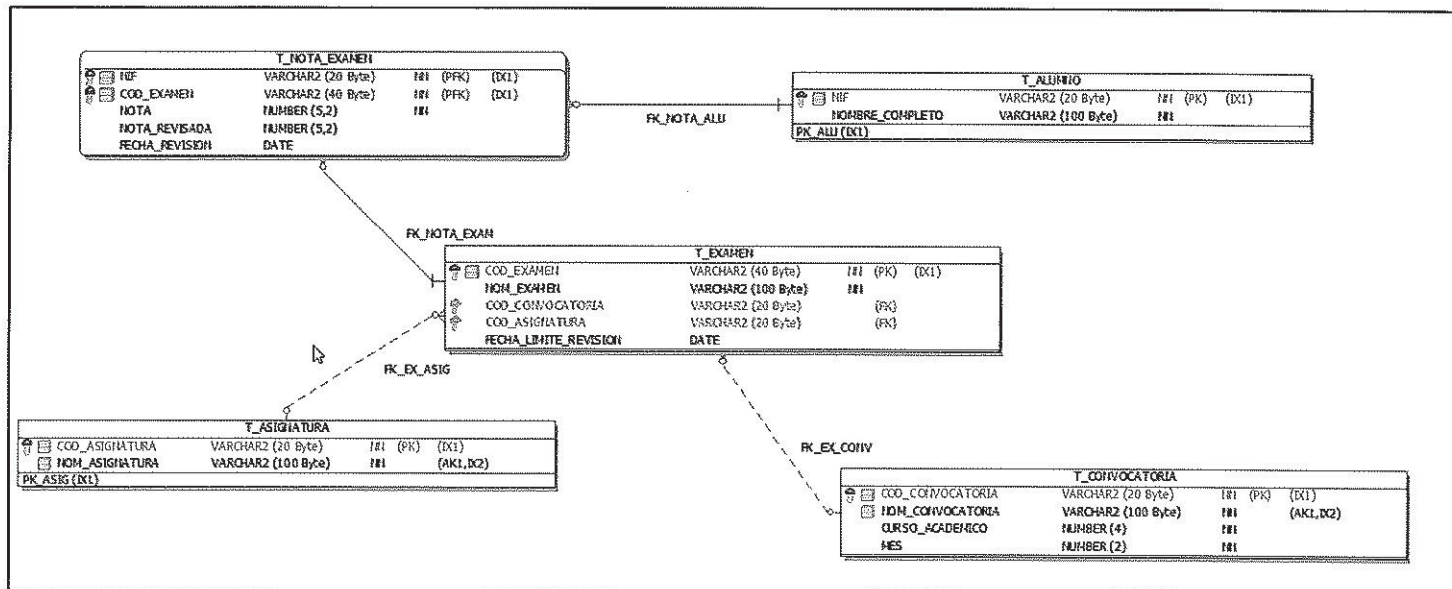
SEGUNDO EJERCICIO

Valencia, 16 de Julio de 2021

PRIMER CASO PRÁCTICO

Enunciado:

Dado el diagrama de tablas de un sistema de gestión notas de exámenes por asignaturas:



Donde:

NN = campo No Nulo

PK = clave primaria de la tabla

FK = clave ajena

PFK = clave primaria formada por claves ajenas

IXn = índice

AKn = constraint unique key

Descripción tablas:

T_ALUMNO : tabla de alumnos, se identifican por campo NIF, debe de tener informado el campo nombre_completo

NIF	NOMBRE_COMPLETO
25415922R	Pepe Pérez
43415988T	Pepe Pérez
43715921W	Andrés López
67898568K	Romualdo García

T_ASIGNATURA: tabla de asignaturas se identifican por el campo cod_asignatura, debe de tener informado el campo nombre_asignatura y no puede repetirse su valor.

COD_ASIGNATURA	NOM_ASIGNATURA
TGRAF_01	Teoría de grafos 01
TGRAF_02	Teoría de grafos 02
AMA_01	Ampliación de matemáticas 01
COMP_01	Compiladores 01
COMP_02	Compiladores 02

T_CONVOCATORIA: tabla de convocatorias se identifica por el campo cod_convocatoria, debe de tener informado el campo nom_convocatoria y no puede repetirse. También debe de tener informado el campo curso_academico que es numérico e identifica el ejercicio en el que finaliza el curso. Así mismo es también obligatorio el campo mes (valor numérico de 1 a 12) que indica el mes en que termina la convocatoria.

COD_CONVOCATORIA	NOM_CONVOCATORIA	CURSO_ACADEMICO	MES
2020_2021_06	Convocatoria junio 2020-2021	2021	7
2020_2021_09	Convocatoria septiembre 2020-2021	2021	9
2021_2022_06	Convocatoria junio 2021-2022	2022	6

T_EXAMEN: tabla de exámenes, un examen se identifica por el campo cod_examen y debe de informar también el campo nombre_examen. Así mismo está relacionado con la convocatoria y la asignatura a la que pertenece el examen.

COD_EXAMEN	NOM_EXAMEN	COD_CONVOCATORIA	COD_ASIGNATURA	FECHA LIMITE_REVISION
2020_2021_06_TGRAF_01	Teoría de grafos 01 convocatoria junio 2020	2020_2021_06	TGRAF_01	
2020_2021_06_AMA_01	Ampliación de matemáticas 01 convocatoria junio 2020	2020_2021_06	AMA_01	

El campo no obligatorio fecha_limite_revision indica la fecha máxima hasta la que un profesor puede revisar las notas del examen.

T_NOTA_EXAMEN: tabla de notas de exámenes en la que se guarda la puntuación de un alumno en un examen. Se identifica por el nif del alumno y el cod_examen del examen. Debe de tener informado el campo nota, y opcionalmente si el profesor revisa el examen con el alumno, puede informarse los campos nota_revisada y fecha_revision.

NIF	COD_EXAMEN	NOTA	NOTA_REVISADA	FECHA_REVISION
25415922R	2020_2021_06_TGRAF_01	9.5		
43715921W	2020_2021_06_TGRAF_01	8	8.3	27/06/2021
67898568K	2020_2021_06_TGRAF_01	8	9.7	28/06/2021

Teniendo en cuenta que el desarrollo se realiza sobre una BD Oracle, resolver las siguientes **preguntas:**

- 1) Crear sentencias DDL de Oracle, de creación de tablas, incluyendo las PK, FK's, índices, restricciones de columnas no nulas, y de campos con valor único. (4 puntos).
- 2) Crear una restricción (constraint) en la tabla T_NOTA_EXAMEN que fuerce a que si se informa el campo nota_revisada, se deba introducir también un valor en el campo fecha_revision, y si el campo fecha_revision está informado también lo debe de estar el campo nota_revisada. (1 punto).
- 3) Crear vista v_max_min_nota_ex , en base a la tabla T_NOTA_EXAMEN que obtenga para todos los puntuaciones de los alumnos en el examen la mínima y máxima nota puntuada, teniendo en cuenta que si está informado el campo nota_revisada se tomará éste como la nota definitiva y sino está informado se tomará el campo nota. (1 punto).

Por ejemplo para el contenido de T_NOTA_EXAMEN :

NIF	COD_EXAMEN	NOTA	NOTA_REVISADA	FECHA_REVISION
25415922R	2020_2021_06_TGRAF_01	9.5		
67898568K	2020_2021_06_TGRAF_01	8	9.7	28/06/2021
43715921W	2020_2021_06_TGRAF_01	8	8.3	27/06/2021

La vista generaría:

NOM_EXAMEN	MIN_NOTA	MAX_NOTA
Teoría de grafos 01 convocatoria junio 2020	8.3	9.7

- 4) Crear vista v_listado_ex_alu, que obtenga un listado, en el que se detalle para cada examen, las personas que se han puntuado con su nota, nota_revisada y fecha de revisión, ordenado por nom_examen, nota ascendente, nota revisada ascendente.

SEGUNDO EJERCICIO DE LAS PRUEBAS SELECTIVAS DE ACCESO AL GRUPO A SUBGRUPO A2
Categoría/Escala Técnico o Técnica Medios de Gestión (Sistemas de Información)

NOM_EXAMEN	NIF	NOMBRE_COMPLETO	NOTA	NOTA_REVISADA	FECHA_REVISION
Ampliación de matemáticas 01 convocatoria junio 2020					
Teoría de grafos 01 convocatoria junio 2020	43715921W	Andrés López	8	8.3	27/06/2021
Teoría de grafos 01 convocatoria junio 2020	25415922R	Pepe Pérez	9.5		

En el caso de que para un examen no se haya puntuado a ningún alumno deberá salir en el listado los campos relacionados con la puntuación y el alumno con valor NULL. (2 puntos).

- 5) Crear trigger en tabla t_nota_examen, que se dispare al crear o actualizar el campo nota_revisada de una tupla y que solo en caso de que el campo nota_revisada esté informado, y el campo fecha_revision no esté informado, informe este último a fecha/hora de sysdate (3 puntos).
- 6) Crear job de BD que se ejecute todos los días a las 6 AM, con el objetivo de que en la tabla t_examen, se informe el campo fecha_limite_revision (sólo si no está informado ya), con la fecha del último día del mes y del año que están guardados respectivamente en los campos mes y curso_academico de la convocatoria asociada al examen.(4 puntos).

SEGUNDO CASO PRÁCTICO

Enunciado:

Una universidad pública ha adquirido un nuevo software para la gestión de todas las becas (transferencias corrientes) que concede, y para la presentación de esta información a los organismos a los que tiene que rendir cuentas según los requisitos que le son impuestos por la legislación vigente.

Este nuevo software se deberá alimentar de los datos existentes en el sistema de información contable para administraciones públicas que ya posee esta universidad. La estructura de datos de este sistema está basada en el estándar del modelo esquemático de funcionamiento de sistemas de contabilidad para administraciones públicas, que es el mismo que por ejemplo se emplea en la UPV.

Teniendo en cuenta que no existe un conector estándar entre ambas piezas de software, y que por tanto se debe crear un exportador de datos del sistema de información contable y un importador de datos para el software de gestión de becas, en este caso práctico nos vamos a centrar en el exportador de datos.

Se ha determinado que el método de interconexión entre ambos sistemas será una nueva tabla en Oracle que contendrá toda la información de las operaciones contables que necesite el nuevo sistema de gestión de becas, y que dicha tabla se irá rellenando con un proceso nocturno que extraiga la información del sistema de información contable de la universidad. Los datos de esta tabla no se borrarán, y se sabrá que ya han sido adquiridos por el sistema de gestión de becas por medio de un campo de la tabla llamado "PROCESADO_SN".

Para la exportación de datos, se procurará incluir toda la información disponible sobre las operaciones contables relacionadas con las becas concedidas, siendo necesario por tanto interrogar a todas las estructuras de datos del sistema de información contable que puedan aportar datos relevantes sobre este tipo de operaciones.

Preguntas

- 1) Definir la estructura de la tabla intermedia que servirá para conectar el nuevo sistema de gestión de becas con el ya existente sistema de información contable. Para ello, escribir la sentencia DDL de Oracle de creación de esta tabla (2 puntos).
- 2) Crear un procedimiento de Oracle que será ejecutado con una programación diaria cada noche para extraer los datos que posteriormente serán importados al nuevo sistema de gestión de becas.
Se valorará que la exportación de datos obtenga la información más completa posible a partir de las estructuras de datos del sistema de información contable.
Este procedimiento se creará empleando las sentencias de Oracle clásicas sin aplicar técnicas de procesamiento masivo (5 puntos).
- 3) Reescribir el procedimiento anterior empleando técnicas de procesamiento masivo BULK COLLECT para mejorar el rendimiento del proceso de exportación. Y escribir las sentencias Oracle necesarias para la creación de un Job que ejecute el procedimiento todas las noches a las 2 de la madrugada. (8 puntos).

TERCER CASO PRÁCTICO

En la Universitat Politècnica de València, entidad sujeta actualmente al Plan General Contable de 2010, y teniendo en cuenta el carácter anual de los créditos, y la desconcentración del gasto por oficinas gestoras y proyectos, al cierre del ejercicio 2021 se plantea los siguientes casos:

- 1) A fecha 31 de diciembre de 2021 en el Dpto. de Termodinámica Aplicada, existen las siguientes situaciones presupuestarias en su oficina gestora:
 - a- Crédito disponible de 780 euros
 - b- Un total de 6.000 euros que corresponden a pedidos pendientes de 2021
 - c- Un expediente de contratación adjudicado por un importe de 10.000 euros
 - d- Las mensualidades de 2021 de los becarios que se incorporaron al departamento en el curso 2021/2022, incluida seguridad social, por un total de 3.600 euros
 - e- Saldo de 2021 del programa de reconocimiento del profesorado de 7.000 euros
 - f- Crédito retenido para la línea presupuestaria del Programa PIME Innovación y mejora educativa por 2.400 euros

- 1.1) Razone con detalle la incorporación o no de cada tipo de saldo y su importe, exponga la justificación normativa según el Presupuesto UPV 2021. (4 puntos).
- 1.2) Respecto al apartado b, exponga si existe algún condicionante para su incorporación o no y explíquelo con detalle. (2 puntos).

- 2) Un PDI del Instituto Universitario de Ingeniería Energética es el responsable económico de un convenio que firmó la UPV en 2021 con la empresa Solar Plak. SL, por un importe total de 1.000.000 de euros, con una duración de tres años. Calcula cual será la incorporación al ejercicio 2022 de este proyecto, teniendo en cuenta estos datos a 31/12/2021:

		Euros
a	Incorporación del ejercicio 2020 al 2021 (corresponde a una factura cobrada el 30/12/2020)	200.000
b	Acumulado de los Derechos reconocidos de facturas en 2021	600.000
c	Acumulado de los ingresos efectivos realizados en 2021	400.000
d	Anticipo de crédito en 2021	100.000
e	Acumulado de las autorizaciones de gastos en 2021	80.000
f	Acumulado de compromiso o disposición de gastos en 2021	75.000
g	Acumulado de reconocimiento de obligaciones en 2021	72.000
h	Acumulado de propuesta de pago en 2021	72.000
i	Acumulado de ordenación del pago en 2021	72.000

SEGUNDO EJERCICIO DE LAS PRUEBAS SELECTIVAS DE ACCESO AL GRUPO A SUBGRUPO A2
Categoría/Escala Técnico o Técnica Medios de Gestión (Sistemas de Información)

j	Acumulado de la realización efectiva del pago en 2021	72.000
k	Acumulado de las Nulas Disponibilidades en 2021 por motivos preventivos	40.000

2.1) Razone la incorporación o no de cada saldo y su importe, exponga la justificación normativa según el Presupuesto UPV 2021 y el Reglamento regulador de las actividades de investigación, desarrollo, transferencia de tecnología y formación no reglada en la UPV. (7 puntos).

2.2) Describa los pasos para hacer dichas incorporaciones, indicando las operaciones presupuestarias a realizar en cada ejercicio. (2 puntos).

CUARTO CASO PRÁCTICO

Enunciado:

Dado un sistema de información contable para administraciones públicas cuyo modelo de datos incluye únicamente las tablas estrictamente necesarias para realizar una operativa básica.

Teniendo en cuenta que en este sistema la tabla de operaciones contables carece de información relativa a la operación encadenada (que normalmente suele indicar el número de la siguiente operación/fase a procesar).

Y que tampoco existe ninguna otra tabla que recoja información relativa a la relación entre las distintas fases del gasto de las operaciones contables del sistema.

Se pretende completar esta operativa básica para añadir al sistema la posibilidad de obtener información relativa a las fases de gasto, con la creación de un nuevo sistema de explotación de tramitaciones del gasto.

En nuestro sistema partimos de las siguientes fases de gestión del gasto que se resumen en los siguientes documentos, sin contemplarse en ningún momento documentos mixtos (aquellos que acumulan dos o más fases):

- Documentos "A"; de autorización de gastos.
- Documentos "D"; de compromiso o disposición de gastos
- Documentos "O"; de reconocimiento de obligaciones
- Documentos "K"; de propuesta de pago
- Documentos "P"; de ordenación del pago
- Documentos "T"; de realización efectiva del pago
- Documentos barrados o negativos de cada una de las fases anteriormente mencionadas, "A/", "D/", "O/", "K/", "P/", "T/".

A través de este nuevo sistema, se persigue el poder conocer la siguiente información:

- A) Dado un número de una operación de autorización de gasto, de compromiso de gasto, o de reconocimiento de la obligación del gasto, saber si se ha realizado el pago efectivo de dicho gasto, aunque sea parcialmente.
- B) Dado un número de una operación, saber si dicha operación ha sido anulada en algún momento de su tramitación mediante la realización de una operación barrada, aunque sea parcialmente.
- C) Dado el número de una operación, obtener el número de la operación inicial que ha originado la tramitación del gasto.
- D) Dado el número de una operación, obtener el árbol completo de la tramitación del gasto, con sus números de operación y tipo de documento de fases de gasto. Utilizar espacios y retornos de carro para representar la jerarquía, y guardar el resultado en una variable VARCHAR2.

- Por ejemplo, en el siguiente caso de árbol de tramitación:

Nº Operación	Tipo/Fase
202100095133	A
202100095136	D
202100242079	O
202100242108	K
202100251938	P
202100254630	T
202100304108	O
202100304109	K
202100317046	P

La variable VARCHAR2 se podría crear de la siguiente forma:

```
v_tram := '202100095133' || chr(13) ||
'202100095136' || chr(32) || 'D' || chr(13) ||
'202100242079' || chr(32) || chr(32) || 'O' || chr(13) ||
'202100242108' || chr(32) || chr(32) || chr(32) || 'K' || chr(13) ||
'202100251938' || chr(32) || chr(32) || chr(32) || chr(32) || 'P' || chr(13) ||
'202100254630' || chr(32) || chr(32) || chr(32) || chr(32) || chr(32) || 'T' || chr(13) ||
'202100304108' || chr(32) || chr(32) || 'O' || chr(13) ||
'202100304109' || chr(32) || chr(32) || chr(32) || 'K' || chr(13) ||
'202100317046' || chr(32) || chr(32) || chr(32) || chr(32) || 'P' || chr(13);
```

siendo el chr(13) el retorno de carro, y el chr(32) el carácter espacio.

- E) Dado el número de una operación, obtener cuál es el número de su operación padre, si la hay.
- F) Dado el número de una operación, obtener cuál es el número de su operación hija, si existe. En el caso de que la operación tenga varias hijas, obtener la más reciente (número de operación más alta).

Preguntas

- 1) En este punto se pide diseñar la estructura de datos necesaria que dé soporte al sistema de explotación de tramitaciones de gasto que se ha expuesto, incluyendo su relación con la tabla de operaciones contables.
Para ello, definir y dibujar el diagrama de tablas, esto debe incluir todos los campos y sus relaciones, las claves primarias y las claves ajenas.
Se valorará la simplicidad y la sencillez, siempre que se cumplan los requisitos enunciados.
(2 puntos).
Para la resolución de este caso práctico, contaremos con una tabla de operaciones contables cuyos campos se limitan únicamente a: número de operación, tipo de fase, descripción e importe.
Los procedimientos de rellenado de estas nuevas tablas no son objeto de este caso práctico, y por lo tanto se considerará que son rellenados de la forma más adecuada por el software según la estructura de datos que se haya definido, en el momento de grabar cada nueva operación en el sistema.
- 2) Escribir las sentencias DDL de Oracle necesarias para la creación de las tablas y de sus relaciones.
(2 puntos).

- 3) Definir la especificación de un paquete de Oracle PL/SQL incluyendo las funciones y procedimientos necesarios para dar soporte a los requisitos de información expuestos en los apartados del A) al F).
(2 puntos).

- 4) Crear el cuerpo del paquete de Oracle PL/SQL especificado en el punto anterior mediante la Implementación de las funciones y procedimientos necesarios para dar soporte a los requisitos de información expuestos en los apartados del A) al C) (4 puntos) y del D) al F) (5 puntos).
En este punto se valorará la eficiencia de la implementación y ejecución mediante el uso de sentencias jerárquicas de Oracle.