

# Universidad Nacional

## Escuela de Informática

Curso: Implementación de bases de datos. (Examen Práctico #2) (11.66% de la nota)

04 de mayo 2018 (8pm) | Profesor: Manuel Espinoza Guerrero.

### **Por favor lea completamente todas las instrucciones antes de iniciar la programación:**

- 1) Se le solicita crear un control para las bonificaciones por empleado. Cada empleado podría registrar varias bonificaciones en años y meses distintos.
- 2) Para cada empleado se pueden registrar varias bonificaciones, el sistema solo maneja 3 tipos de bonificaciones:

a. Ventas	<b>Porcentaje indicado% Solo para Vendedores</b>
b. Evaluación	<b>1.00% Fijo para todos los empleados</b>
c. Gerencia	Se registra un monto manualmente
- 3) Para cada empleado se registra solo una bonificación por tipo, año y mes de salario. Para cada bonificación se registra siempre un monto. Para el registro del monto de cada bonificación el campo debe ser de 2 decimales. Es decir, debe ser number(8,2).
- 4) Se debe crear un procedimiento en BD, que se indique un Empleado, un año, un mes y un monto de venta, además opcionalmente puede indicarse un monto adicional que correspondería al monto de bonificación de Gerencia, si no se indica dicho monto significa que para dicho empleado no se debe registrar un monto de bonificación de Gerencia.
- 5) Se sabe que un empleado es vendedor cuando el campo COMMISSION\_PCT no es nulo, para los empleados que son vendedores deben multiplicar el Monto de las Ventas x COMMISSION\_PCT (divido entre 100).
- 6) El procedimiento debe tomar el monto de las ventas del empleado y realizar los cálculos según el porcentaje de comisión (cuando no es nulo), si es nulo se debe insertar cero.
- 7) Si el procedimiento se ejecuta dos veces para el mismo empleado con un mismo mes y año, deberá generar un error DEFAULT (Error de PK o UK) ya que debería estar quebrantando una regla de integridad de la base de datos, no debe presentar mensaje personalizado, ni excepción personalizada.
- 8) Crear una función que sume todas las bonificaciones de un empleado. Usar esta función en la vista que se solicita.
- 9) Debe crear una vista, que muestre id empleado, email, salario, total de bonificaciones, total del pago (salario más total de bonificaciones) solo para los empleados que se les ha calculado bonificaciones. Ordenado por el salario de mayor a menor.
- 10) Debe crear un trigger que, al momento de insertar bonificaciones que estén con monto en negativo, las registre con monto CERO, siempre, y no debe generar error solo sustitución.
- 11) Debe hacer un select de la vista para mostrar los resultados con formato adecuado, además de dos selects adicionales solicitados en el proceso.

# Universidad Nacional

## Escuela de Informática

Curso: Implementación de bases de datos. (Examen Práctico #2) (11.66% de la nota)

04 de mayo 2018 (8pm) | Profesor: Manuel Espinoza Guerrero.

### Proceso:

Luego de creados los objetos (recursivo, sin errores)

Debe ejecutar el Procedimiento para procesar las bonificaciones de Mayo del 2018 para estos 4 empleados:

ID	Apellido	Salario	Ventas	PCT Comm	Evaluacion 1%	Monto Comm Ventas	Monto Evaluación	Monto Gerencia	Tot Bonif	A pagar	Cant Reg.
168	Ozer	11,500.00	50,000.00	0.25	1.00	125.00	115.00	550.00	790.00	12,290.00	3
103	Hunold	9,000.00	0.00	null	1.00	0.00	90.00	0.00	90.00	9,090.00	3
160	Doran	7,500.00	100,000.00	0.30	1.00	300.00	75.00	125.00	500.00	8,000.00	3
105	Austin	4,800.00	12,000.00	null	1.00	0.00	48.00	0.00	48.00	4,848.00	3
Total Ded:										12	

El procedimiento se debe ejecutar 5 veces (1 error esperado) Con los parámetros: **Empleado, Año, Mes, Monto de Ventas, "Monto Gerencia" (4 o 5 parámetros el mismo procedimiento):**

```
PROMPT #1 Ventas $50mil, monto de gerencia $550
execute prc_calc_bonif(168,2018,5,50000,550);

PROMPT #2 Ventas Cero, monto gerencia $-50 se pasa a cero, no da error
execute prc_calc_bonif(103,2018,5,0,-50);

PROMPT #3 Ventas por $100,000 monto gerencia por $125
execute prc_calc_bonif(160,2018,5,100000,125);

PROMPT #4 Monto ventas $12000, pero no gana comision, monto gerencia no se indica, pone cero
execute prc_calc_bonif(105,2018,5,12000);

PROMPT #5 Error esperado de PK o UK, mismo cálculo que el anterior
execute prc_calc_bonif(105,2018,5,12000);
```

Y deben mostrarse los siguientes mensajes en la salida del script (solo un error esperado, PK o UK):

```
#1 Ventas $50mil, monto de gerencia $550
PL/SQL procedure successfully completed.
#2 Ventas Cero, monto gerencia $-50 se pasa a cero, no da error
PL/SQL procedure successfully completed.
#3 Ventas por $100,000 monto gerencia por $125
PL/SQL procedure successfully completed.
#4 Monto ventas $12000, pero no gana comision, monto gerencia no se indica, pone
cero
PL/SQL procedure successfully completed.
#5 Error esperado de PK o UK, mismo cálculo que el anterior
BEGIN prc_calc_bonif(105,2018,5,12000); END;

*
ERROR at line 1:
ORA-00001: unique constraint (HR.BONIFICA_PK) violated
ORA-06512: at "HR.PRC_CALC_BONIF", line 11
ORA-06512: at line 1
```

## Universidad Nacional

Escuela de Informática

Curso: Implementación de bases de datos. (Examen Práctico #2) (11.66% de la nota)

04 de mayo 2018 (8pm) | Profesor: Manuel Espinoza Guerrero.

Con la ejecución 4 veces de forma correcta (Nota: que uno da error, entonces no se registra), debe hacer un conteo de cantidad de bonificaciones que se registraron en cero (son 4 registros), el total de registros de la tabla debe ser 12 registros, pues si tiene monto cero en bonificación igual se debe insertar:

```
Cantidad de Registros con monto cero
CANT_MON_CERO
-----
                4

Total de Registros en Tabla Bonificaciones
CANT_DEDUC
-----
                12
```

Esta es la salida de la consulta de la vista, debe ordenarse por salario:

```
Reporte de la vista
EMP_ID  APELLIDO      SALARIO      TOT_BON      TOT_PAGO
-----
    168  Ozer         11,500.00      790.00    12,290.00
    103  Hunold        9,000.00       90.00     9,090.00
    160  Doran        7,500.00      500.00     8,000.00
    105  Austin        4,800.00       48.00     4,848.00
```

Nota: Use el siguiente formato, para las columnas relacionadas con salario, total de bonificaciones y total de pago que se usa en la consulta de la vista:

`·format·99,999.00`

# Universidad Nacional

## Escuela de Informática

Curso: Implementación de bases de datos. (Examen Práctico #2) (11.66% de la nota)

04 de mayo 2018 (8pm) | Profesor: Manuel Espinoza Guerrero.

### Consideraciones y Puntaje:

- 1) Crear los nombres de los objetos de acuerdo con las buenas prácticas.
- 2) Script totalmente recursivo, solo un error esperado en la ejecución del procedimiento.
- 3) Documento con PROMPT cada sección importante del script.
- 4) Recuerde el orden del script, según lo visto en clases.

<b>Puntaje:</b>
<b>5ptos</b>
<b>10ptos</b>
<b>35ptos</b>
<b>10ptos</b>
<b>5ptos</b>
<b>15ptos</b>
<b>10ptos</b>
<b>5ptos</b>
<b>5ptos</b>
- a. Drop de objetos independientes
- b. Tabla con PK, FK, CK, UK, null o not null
- c. Procedimiento que calcula bonificaciones
- d. Función de sumar bonificaciones por empleado
- e. Crear trigger
- f. Crear vista
- g. Dos selects solicitados
- h. Reporte de vista bonificaciones
- i. Nomenclatura correcta para nombres de objetos
- 5) Debe crear todas las restricciones de integridad que cumpla de forma óptima las reglas de negocio indicadas (PK's, FK's CK's, UK's) y null, not null según se solicita.
- 6) Coloque al final del script su nombre completo en un PROMPT.
- 7) Ejecutar los procedimientos exactamente con los valores y empleados indicados en los ejemplos, ni más, ni menos. Debe mostrarse el "error esperado". Pero sí puede crear funciones adicionales si así lo considera apropiado.
- 8) Para los reportes de las vistas y los dos selects adicionales debe respetar el orden de los registros.
- 9) Hacer todo para el usuario system/root, que éste se conecte con hr para crear los objetos solicitados (como lo practicamos muchas veces en clases)
- 10) El sistema contempla la creación solamente de una tabla.
- 11) Luego de creados los objetos debe:
  - a. Ejecutar el procedimiento 5 veces (1 error esperado)
  - b. Hacer los dos selects solicitados
  - c. Hacer un select de la vista
- 12) Los resultados esperados fueron mostrados en los ejemplos anteriores.

**Entregable:** Script recursivo con prefijo "ep2\_" y su nombre\_apellido.sql subido en el Aula Virtual. Sin comprimir, solo el archivo .SQL de la práctica, verificar que sea el archivo que corresponde.

Ejemplo de nombre de archivo: **ep2\_manuel\_espinoza.sql**

ep2 = examen práctico #2