

UNERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

INFORME DE LABORATORIO No 02

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Cuadros Quiroga

Integrantes:

Renzo A. Moreno Cáceres	(2013047246)
Mamani Limache, Jhony	(2013046566)
Nombre Estudiante No 3	(Codigo 03)
Nombre Estudiante No 4	(Codigo 04)
Nombre Estudiante No 5	(Codigo 05)

Índice

1. Actividad No 01 – Manipulación de Datos

1. El departamento de Recursos Humanos requiere crear sentencias SQL para insertar, actualizar y eliminar datos de empleados. Como prueba se utilizará la tabla Mis_Empleados antes de remitir las sentencias al departamento de Recursos Humanos.
2. Crear la tabla Mis_Empleados utilizando la siguiente estructura.

Atributo	Tipo de Dato	Permite Nulos
ID	NUMBER(4)	NO
APELLIDOS	VARCHAR2(25)	SI
NOMBRES	VARCHAR2(25)	SI
CODIGO	VARCHAR2(10)	SI
SALARIO	NUMBER(9,2)	SI

```
10 --Ejercicio 2
11 create table mis_empleados (
12     empleados_id int not null,
13     apellidos varchar(25),
14     nombres varchar(25),
15     codigo varchar(10),
16     salario decimal(9,2)
17 )
18 go
```

3. Generar una sentencia de inserción de datos que permita añadir los siguientes registros:

ID	APELLIDOS	NOMBRES	CODIGO	SALARIO
1	Vargas Canseco	Raúl	Rvargas	895
2	Castro Feria	María	mcastro	860

empleados_id	apellidos	nombres	codigo	salario
1	Vargas Canseco	Raúl	rvargas	895.00
2	Castro Feria	María	mcastro	860.00

4. Generar un script que permita que mediante utilización de variables de sustitución, la inserción de información en la tabla Mis_Empleados.

```
27 --Ejercicio 4
28 create proc insertar_datos (@id int, @ape varchar(25), @nom varchar(25), @cod varchar(10), @sal decimal(9,2))
29 as
30 begin
31     insert into mis_empleados values (@id,@ape,@nom,@cod,@sal)
32 end
33 go
```

5. Utilizando el script anterior adicionar los siguientes registros.

ID	APELLIDOS	NOMBRES	CODIGO	SALARIO
3	Gómez Albán	Juan Pablo	Jgomez	1100
4	Quiroz Ardiles	Judith	Jquiroz	750
5	Soria Peralta	Pedro	Psoria	1550

```
34 --Ejercicio 5
35 exec insertar_datos 3,'Gómez Albán','Juan Pablo','Jgomez',1100;
36 exec insertar_datos 4,'Quiroz Ardiles','Judith','Jquiroz',750;
37 exec insertar_datos 5,'Soria Peralta','Pedro','Psoria',1550;
38 go
```

6. Revisar los cambios hechos a la tabla.

empleados_id	apellidos	nombres	codigo	salario
1	Vargas Canseco	Raúl	rvargas	895.00
2	Castro Feria	María	mcastro	860.00
3	Gómez Albán	Juan Pablo	Jgomez	1100.00
4	Quiroz Ardiles	Judith	Jquiroz	750.00
5	Soria Peralta	Pedro	psoria	1550.00

7. Cambiar el nombre del empleado No 3 a Benjamín.

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	895.00
2	2	Castro Feria	María	mcastro	860.00
3	3	Gómez Albán	Benjamin	lgomez	1100.00
4	4	Quiroz Ardies	Judith	jquiroz	750.00
5	5	Soria Peralta	Pedro	psoria	1550.00

8. Elevar el salario a \$ 1,000 a todos los empleados que tengan un salario menor a esa cantidad.

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Casaseco	Raúl	rvargas	1000.00
2	2	Castro Feria	María	mcastro	1000.00
3	3	Gómez Albalá	Benjamín	igomez	1100.00
4	4	Quiroz Ardiles	Judith	jquiroz	1000.00
5	5	Soria Peralta	Pedro	psoria	1550.00

9. Eliminar el registro del empleado María Castro

```
--
57 --Ejercicio 9
58 delete from mis_empleados where codigo='mcastro';
59 go
60
```

121 % Mensajes

	empleados_id	apellidos	nombrs	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	1000.00
2	2	Gómez Albin	Benjamín	gomez	1100.00
3	4	Queroz Andres	Juanro	juaroz	1000.00
4	5	Soria Penilla	Pedro	psoria	1550.00

10. Revisar los cambios hechos a la tabla.

```
61 --ejercicio 10
62 select [Begin Time],[RowLop Contents],[Transaction Name],Operation
63 from fn_log('SQLSERVER',NULL,NULL)
64 where AllocInitName='dbo.us_empleados' and Operation IN ('LOP_DELETE_ROWS')
65 go
```

22 %

	Results	Msg
1	Begin Time	RowLop Contents Transaction Name Operation
2	NUL	9d4f108bc38004+10013680000+10200442030004 NUL LOP_DELETE_ROWS
3	NUL	9d4f108bc38004+10013680000+10200442030004 NUL LOP_DELETE_ROWS

11. Confirmar los cambios a la tabla.

```
68
69 --Ejercicio 11
70 select * from mis_empleados
71 go
72
```

121 %

File Windows Messages

Begin Time	RowLog Contents 1	Transaction Name	Operation
1 NULL	(a0101000c0000410101360000102000410200004	NULL	LOG_DELETE_ROWS
2 NULL	(a0101000c0000410101360000102000410200004	NULL	LOG_DELETE_ROWS

12. Adicionar el siguiente registro a la tabla

13. Revisar la adición realizada

14. Crear un punto de restauración intermedio para esta transacción

15. Borrar los registros de la tabla MIS_EMPLEADOS.

16. Revisar los cambios realizados.

17. Descartar los cambios hechos a la tabla sin descartar la última adición hecha.

18. Revisar nuevamente los registros de la tabla MIS_EMPLEADOS.

19. Confirmar todos los cambios hechos a la tabla MIS_EMPLEADOS.
20. Modificar el script del punto 4.4. a fin de que se genere automáticamente el CODIGO del empleado que lo conforman la primera letra de su nombre y la primera palabra de su apellido.
21. Adicionar el siguiente registro a la tabla a fin de corroborar el funcionamiento del script anterior
22. Revisar los cambios realizados. Y finalmente confirmar todos los cambios hechos a la tabla MIS_EMPLEADOS.

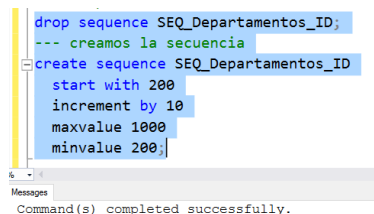
2. Actividad No 02 – Reconociendo la estructura

1. Crear la tabla Departamentos utilizando la siguiente estructura:
2. Poblar la tabla Departamentos con los datos de la tabla Departments.
3. Crear la tabla Empleados utilizando la siguiente estructura.
4. Crear la tabla Empleados2 basada en la estructura de la tabla Employees. Incluir solo las columnas EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY y DEPARTMENT_ID respectivamente.
5. Modificar el estado de la tabla Empleados2 a SOLO LECTURA.
6. Tratar de adicionar el siguiente registro a la tabla Empleados2.
7. Revertir el estado de la tabla LECTURA / ESCRITURA. Tratar de insertar nuevamente la información del punto 4.6.
8. Eliminar la tabla Empleados2.

3. Actividad No 03 – Otros objetos de base de datos

1. El Departamento de Recursos Humanos requiere ocultar ciertos datos de la tabla EMPLOYEES, Ellos necesitan una vista llamada VW_Empleados, que contenga los campos ID del Empleado, Nombres e ID del Departamento.
2. Utilizando la vista anterior crear un reporte que muestre los nombres y departamentos a los cuales pertenecen los empleados.
3. El departamento 50 requiere acceso a los datos de los empleados. Generar una vista llamada VW_Dept50, que contenga las columnas ID del Empleado, Apellidos e ID del Departamento de los empleados del departamento 50. Etiquetar las columnas como EmpNo, Empleado y DeptNo. Por razones de seguridad no se debe permitir a los empleados ser reasignados a otros departamentos.
4. Probar la vista, tratando de reasignar al empleado Matos al departamento 80.
5. Se requiere crear una secuencia que será utilizada en la Llave Primaria de la tabla Departamentos (tabla creada en la práctica anterior). La secuencia deberá iniciar con el valor 200 y terminar en el valor 1000, asimismo deberá incrementarse en 10 cada vez que se requiera. Nombrar la secuencia SEQ_Departamentos_ID.

```
create sequence SEQ_Departamentos_ID start with 200 increment by 10 maxvalue 1000 minvalue 200;
```



```
drop sequence SEQ_Departamentos_ID;
-- creamos la secuencia
create sequence SEQ_Departamentos_ID
start with 200
increment by 10
maxvalue 1000
minvalue 200;
```

Messages
Command(s) completed successfully.

6. Para probar la secuencia, adicionar dos registros a la tabla Departamentos, Educación y Administración. Verificar la adición.

```
declare @liCodigo int select @liCodigo = next value for SEQ_Departamentos_ID insert into departments values(@liCodigo,'matematica','300','3300') select * from departments
```

	department_id	department_name	manager_id	location_id
1	2	educacion	300	3300
2	10	Administration	200	1700
3	20	Marketing	201	1800
4	30	Purchasing	114	1700
5	40	Human Resources	203	2400
6	50	Shipping	121	1500
7	60	IT	103	1400
8	70	Public Relations	204	2700
9	80	Sales	145	2500
10	90	Executive	100	1700
11	100	Finance	108	1700
12	110	Accounting	205	1700
13	120	Treasury	NULL	1700
14	130	Corporate Tax	NULL	1700
15	140	Control And Credit	NULL	1700
16	150	Shareholder Services	NULL	1700
17	160	Benefits	NULL	1700
18	170	Manufacturing	NULL	1700
19	180	Construction	NULL	1700
20	190	Contracting	NULL	1700
21	200	Operations	NULL	1700
22	210	IT Support	NULL	1700
23	220	NOC	NULL	1700
24	230	IT Helpdesk	NULL	1700
25	240	Government Sales	NULL	1700
26	250	Retail Sales	NULL	1700
27	260	Recruiting	NULL	1700
28	270	Payroll	NULL	1700
29	300	educacion	300	3300
30	370	educacion	300	3300
31	380	matematica	300	3300

7. Crear un índice no único en la columna NOMBRE de la tabla Departamentos.

```
CREATE INDEX Indice_no_unico ON departments (department_name);
```

```

select * from departments
CREATE INDEX Indice_no_unico
ON departments (department_name);

```

Messages
Command(s) completed successfully.

8. Crear un sinónimo para la tabla EMPLOYEES con el nombre EMP.

```
EXECUTE sp_addlinkedserver Server1; GO CREATE SYNONYM EMP FOR Server1.AdventureWorks2012.HumanResources.Employee; GO
```

```

EXECUTE sp_addlinkedserver Server1;
GO
CREATE SYNONYM EMP
FOR Server1.AdventureWorks2012.HumanResources.Employee;
GO

```

Messages
Command(s) completed successfully.