

UNIVERSIDAD 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)
Ordoñez Quilli, Ronald	(2015052821)
Nombre Estudiante No 4	(Codigo 04)
Nombre Estudiante No 5	(Codigo 05)

Índice

1. Actividad No 01 – Manipulación de Datos	1
2. Actividad No 02 – Reconociendo la estructura	8
3. Actividad No 03 – Otros objetos de base de datos	9

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:

```
insert into mis_empleados values (1, 'Vargas Canseco','Raúl', 'rvargas', 895),(2, 'Castro Feria',
'María','mcastro', 860);
go
```

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

Resultados

Mensajes

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	895.00
2	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
28  --Ejercicio 4
29  create proc insertar_datos @id int, @ape varchar(25), @nom varchar(25), @cod varchar(10), @sal decimal(9,2)
30  as
31  insert into mis_empleados values (@id,@ape,@nom,@cod,@sal);
32  go
33
34
```

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
35  --Ejercicio 5
36  exec insertar_datos 3, 'Gómez Albán', 'Juan Pablo', 'jgomez', 1100;
37  exec insertar_datos 4, 'Quiroz Ardiles', 'Judith', 'jquiroz', 750;
38  exec insertar_datos 5, 'Soria Peralta', 'Pedro', 'psoria', 1550;
39  go
40
```

6. Revisar los cambios hechos a la tabla.

```
select * from mis_empleados
go
```

Resultados		Mensajes			
	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	Juan Pablo	jgomez	1100.00
4	4	Quiroz Ardiles	Judith	jquiroz	750.00
5	5	Soria Peralta	Pedro	psoria	1550.00

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

```
update mis_empleados set nombres='Benjamin' where empleados_id=3;
go
```

Resultados		Mensajes			
	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	jgomez	1100.00
4	4	Quiroz Ardiles	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.

```
update mis_empleados set salario=1000 where salario<1000;
go
```

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	1000.00
2	2	Castro Feria	María	mcastro	1000.00
3	3	Gómez Albán	Benjamin	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

```
delete from mis_empleados where codigo='mcastro';
go
```

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

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	1000.00
2	3	Gómez Albán	Benjamin	igomez	1100.00
3	4	Quiroz Ardiles	Judith	jquiroz	1000.00
4	5	Soria Peralta	Pedro	psoria	1550.00

10. Revisar los cambios hechos a la tabla.

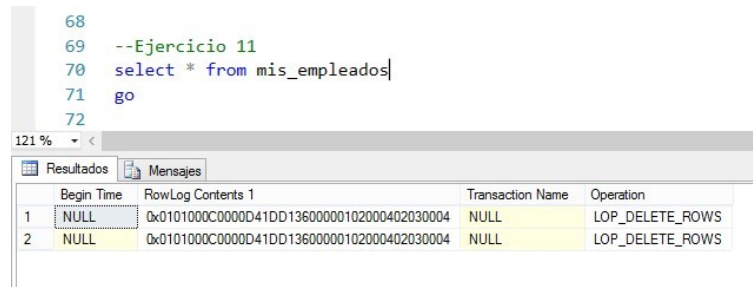
```
select [Begin Time],[RowLog Contents 1],[Transaction Name],Operation
from sys.fn_dblog(NULL,NULL)
where AllocUnitName='dbo.mis_empleados' and Operation IN ('LOP_DELETE_ROWS')
go
```

```
61
62  --Ejercicio 10
63  select [Begin Time],[RowLog Contents 1],[Transaction Name],Operation
64  from sys.fn_dblog(NULL,NULL)
65  where AllocUnitName='dbo.mis_empleados' and Operation IN ('LOP_DELETE_ROWS')
66  go
67
```

	Begin Time	RowLog Contents 1	Transaction Name	Operation
1	NULL	0x0101000C0000D41DD13600000102000402030004	NULL	LOP_DELETE_ROWS
2	NULL	0x0101000C0000D41DD13600000102000402030004	NULL	LOP_DELETE_ROWS

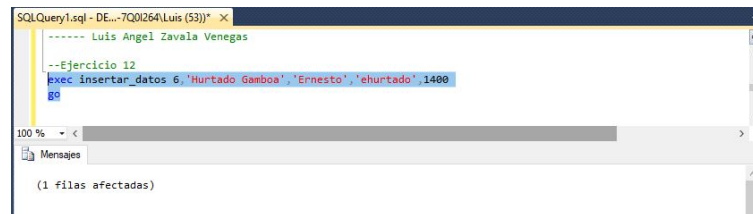
11. Confirmar los cambios a la tabla.

```
select * from mis_empleados  
go
```



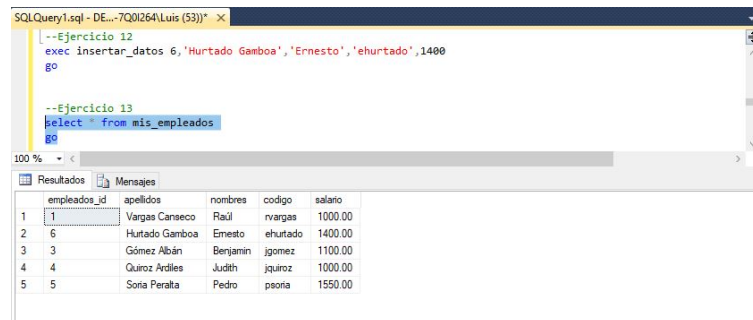
12. Adicionar el siguiente registro a la tabla

```
exec insertar_datos 6,'Hurtado Gamboa','Ernesto','ehurtado',1400  
go
```



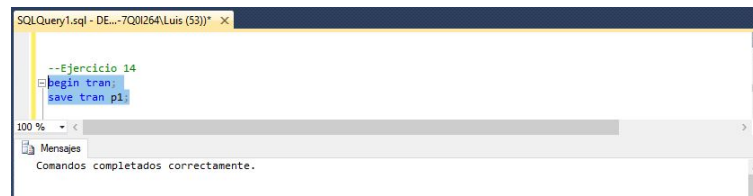
13. Revisar la adición realizada

```
select * from mis_empleados  
go
```



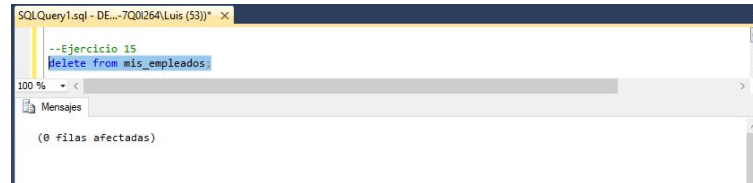
14. Crear un punto de restauración intermedio para esta transacción

```
begin tran;  
save tran p1;
```



15. Borrar los registros de la tabla MIS_EMPLEADOS.

```
delete from mis_empleados;
```



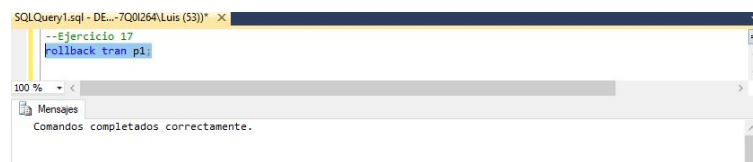
16. Revisar los cambios realizados.

```
select * from mis_empleados;
```



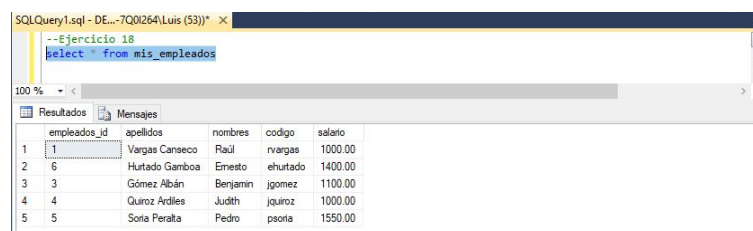
17. Descartar los cambios hechos a la tabla sin descartar la última adición hecha.

```
rollback tran p1;
```



18. Revisar nuevamente los registros de la tabla MIS_EMPLEADOS.

```
select * from mis_empleados
```



19. Confirmar todos los cambios hechos a la tabla MIS_EMPLEADOS.

```
select [Begin Time],[RowLog Contents 1],[Transaction Name],Operation
from sys.fn_dblog(NULL,NULL)
where AllocUnitName='dbo.mis_empleados' and Operation IN ('LOP_DELETE_ROWS')
go
```

	Begin Time	RowLog Contents 1	Transaction Name	Operation
1	NULL	0x0101000C0000D1D9C83400000102000402030004	NULL	LOP_DELETE_ROWS
2	NULL	0x0101000C0000D1D9C83400000102000402030004	NULL	LOP_DELETE_ROWS
3	NULL	0x0101000C0000D1D9C83400000102000402030004	NULL	LOP_DELETE_ROWS
4	NULL	0x0101000C0000D1D9C83400000102000402030004	NULL	LOP_DELETE_ROWS
5	NULL	0x0101000C0000D1D9C83400000102000402030004	NULL	LOP_DELETE_ROWS

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.

```
alter proc insertar_datos @id int, @ape varchar(25), @nom varchar(25), @sal decimal(9,2)
as
declare @cod varchar(10);
set @cod = LEFT(@nom,1) + @ape;
insert into mis_empleados values (@id,@ape,@nom,@cod,@sal);
go
```

Comandos completados correctamente.

21. Adicionar el siguiente registro a la tabla a fin de corroborar el funcionamiento del script anterior

```
exec insertar_datos 7,'Valdivia Pérez','Graciela',1800;
```

go

(1 filas afectadas)

22. Revisar los cambios realizados. Y finalmente confirmar todos los cambios hechos a la tabla MIS_EMPLEADOS.

```
select * from mis_empleados
select [Current LSN],[Transaction ID],[RowLog Contents 0],Operation
from sys.fn_dblog(NULL,NULL)
where AllocUnitName='dbo.mis_empleados' and Operation IN ('LOP_INSERT_ROWS')
go
```


SQLQuery1.sql - DE...-7Q0I264/Luis (53)*

```
--Ejercicio 22
select * from mis_empleados
select [Current LSN],[Transaction ID],[RowLog Contents 0],[Operation]
from sys.fn_dblog(NULL,NULL)
where AllocUnitName='dbo.mis_empleados' and Operation IN ('LOP_INSERT_ROWS')
go
```

100 %

Resultados Mensajes

	empleados_id	apellidos	nombres	codigo	salario
1	1	Vargas Canseco	Raúl	rvargas	1000.00
2	6	Hurtado Gamboa	Ernesto	ehurtado	1400.00
3	3	Gómez Albán	Benjamin	lgomez	1100.00
4	4	Quiroz Aciles	Judith	jquiroz	1000.00
5	5	Soria Peralta	Pedro	psoria	1550.00
6	7	Valdivia Pérez	Graciela	GValdivia	1800.00

	Current LSN	Transaction ID	RowLog Contents 0	Operation
1	00000023:00000710:001a	0000:000003d5	0x30000D0001000000019C5D0100050000030026002A00310...	LOP_INSERT_ROWS
2	00000023:00000710:001c	0000:000003d5	0x30000D000200000001F04F0100050000030024002900300...	LOP_INSERT_ROWS
3	00000023:00000728:0002	0000:000003d5	0x30000D000300000001B0AD0100050000030023002C00330...	LOP_INSERT_ROWS
4	00000023:00000730:0002	0000:000003d5	0x30000D000400000001F8240100050000030026002C00330...	LOP_INSERT_ROWS
5	00000023:00000738:0002	0000:000003d5	0x30000D000500000001785D0200050000030025002A00350...	LOP_INSERT_ROWS
6	00000023:00000778:0002	0000:000003e7	0x30000D000600000001E0220200050000030026002D00350...	LOP_INSERT_ROWS
7	00000023:00000780:000a	0000:000003e8	0x30000D000500000001785D0200050000030025002A00350...	LOP_INSERT_ROWS
8	00000023:00000780:000b	0000:000003e8	0x30000D000400000001A0860100050000030026002C00330...	LOP_INSERT_ROWS
9	00000023:00000780:000c	0000:000003e8	0x30000D000300000001B0AD0100050000030023002C00331...	LOP_INSERT_ROWS
10	00000023:00000780:000d	0000:000003e8	0x30000D000600000001E0220200050000030026002D00350...	LOP_INSERT_ROWS

Consulta ejecutada correctamente. DESKTOP-7Q0I264 (12.0 SP1) DESKTOP-7Q0I264/Luis (53) Human_Resources 00:00:00 18 filas

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.

```
CREATE VIEW VW_Empleados AS
SELECT employee_id, last_name employee, department_id
FROM employees;
GO
```



- Utilizando la vista anterior crear un reporte que muestre los nombres y departamentos a los cuales pertenecen los empleados.

```
SELECT employee, department_id
FROM VW_Empleados;
```

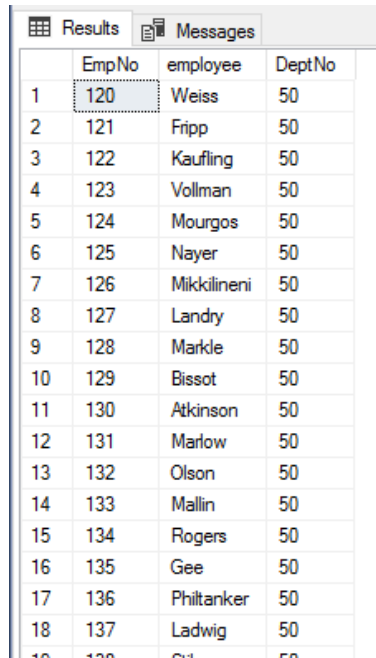
	employee	department_id
1	King	90
2	Kochhar	90
3	De Haan	90
4	Hunold	60
5	Ernst	60
6	Austin	60
7	Pataballa	60
8	Lorentz	60
9	Greenberg	100
10	Faviet	100
11	Chen	100
12	Sciarra	100
13	Uman	100
14	Popp	100
15	Raphaely	30
16	Khoo	30
17	Baida	30
18	Tobias	30
19	Himuro	30

Query executed successfully.

- 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.

```
CREATE VIEW VW_Dept50 AS
SELECT employee_id EmpNo, last_name employee, department_id DeptNo
FROM employees
```

```
WHERE department_id = 50;  
GO
```

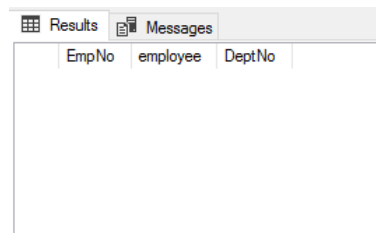


The screenshot shows a SQL Server query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with four columns: 'EmpNo', 'employee', and 'DeptNo'. The table contains 18 rows of data, with the first row highlighted. The 'EmpNo' column ranges from 120 to 137, and the 'DeptNo' column is consistently 50.

	EmpNo	employee	DeptNo
1	120	Weiss	50
2	121	Fripp	50
3	122	Kaufling	50
4	123	Vollman	50
5	124	Mourgos	50
6	125	Nayer	50
7	126	Mikkilineni	50
8	127	Landry	50
9	128	Markle	50
10	129	Bissot	50
11	130	Atkinson	50
12	131	Marlow	50
13	132	Olson	50
14	133	Mallin	50
15	134	Rogers	50
16	135	Gee	50
17	136	Philtanker	50
18	137	Ladwig	50

4. Probar la vista, tratando de reasignar al empleado Matos al departamento 80.

```
UPDATE VW_Dept50  
SET DeptNo = 80;  
GO
```



The screenshot shows a SQL Server query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with three columns: 'EmpNo', 'employee', and 'DeptNo'. The table is currently empty.

EmpNo	employee	DeptNo
-------	----------	--------

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.

- 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

- 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.

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

EXECUTE sp_addlinkedserver Server1; GO CREATE SYNONYM EMP FOR Server1.AdventureWorks201
GO

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

Messages
Command(s) completed successfully.