

UNERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

INFORME DE LABORATORIO No 01

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Cuadros Quiroga

Integrantes:

Balaguer Valle Angela	(2016054494)
Huallpa Catro Leydi	(2015053230)
Mamani Ayala Brandon	(2015052715)
Pilco Quispe Mireya	(2015053234)
Quispe Mamani Angelo	(2015052826)
Vizcarra Llanque Jhordy	(2015052719)

Índice

1. Actividad No 01 – Revisión de Sintaxis	1
2. Actividad No 02 – Reconociendo la estructura	3
3. Actividad No 03 – Consultas Básicas	5
4. Actividad No 04 – Restricción y Ordenamiento	8
5. Actividad No 05 – Funciones	15
6. Actividad No 06 – Funciones de Conversión	18
7. Actividad No 07 – Funciones de Agrupación	21
8. Actividad No 08 – Enlaces	23
9. Actividad No 09 – SubConsultas	27
10. Actividad No 10 – Conjuntos	32

1. Actividad No 01 – Revisión de Sintaxis

De los siguientes comandos ¿Cuál es el resultado? ¿En caso de ser error cual sería la sentencia correcta?

- SELECT last_name, job_id, salary AS Sal FROM employees;
Es correcta

	last_name	job_id	Sal
1	King	AD_PRES	24000.00
2	Kochhar	AD_VP	17000.00
3	De Haan	AD_VP	17000.00
4	Hunold	IT_PROG	9000.00
5	Ernst	IT_PROG	6000.00
6	Austin	IT_PROG	4800.00
7	Pataballa	IT_PROG	4800.00
8	Lorentz	IT_PROG	4200.00
9	Greenberg	FI_MGR	12008.00
10	Faviet	FI_ACC...	9000.00
11	Chen	FI_ACC...	8200.00
12	Sciarra	FI_ACC...	7700.00
13	Urman	FI_ACC...	7800.00
14	Popp	FI_ACC...	6900.00
15	Raphaely	PU_MAN	11000.00
16	Khoo	PU_CLE...	3100.00
17	Baida	PU_CLE...	2900.00
18	Tobias	PU_CLE...	2800.00
19	Himuro	PU_CLE...	2600.00

- SELECT * FROM job_grades;
Es incorrecta, la sentencia correcta sería:
SELECT * FROM jobs;

	job_id	job_title	min_salary	max_salary
1	AC_ACCOUNT	Public Accountant	4200	9000
2	AC_MGR	Accounting Manager	8200	16000
3	AD_ASST	Administration Assistant	3000	6000
4	AD_PRES	President	20080	40000
5	AD_VP	Administration Vice Pr...	15000	30000
6	FI_ACCOUNT	Accountant	4200	9000
7	FI_MGR	Finance Manager	8200	16000
8	HR_REP	Human Resources R...	4000	9000
9	IT_PROG	Programmer	4000	10000
10	MK_MAN	Marketing Manager	9000	15000
11	MK_REP	Marketing Represent...	4000	9000

- SELECT employee_id, last_name sal x 12 ANNUAL SALARY FROM employees;
Es incorrecta, la sentencia correcta sería:
SELECT employee_id, last_name, salary * 12 'ANNUAL SALARY' FROM employees;

	employee_id	last_name	ANNUAL SALARY
1	100	King	288000.00
2	101	Kochhar	204000.00
3	102	De Haan	204000.00
4	103	Hunold	108000.00
5	104	Ernst	72000.00
6	105	Austin	57600.00
7	106	Pataballa	57600.00
8	107	Lorentz	50400.00
9	108	Greenb...	144096.00
10	109	Faviet	108000.00
11	110	Chen	98400.00
12	111	Sciarra	92400.00
13	112	Uman	93600.00
14	113	Popp	82800.00
15	114	Raphaely	132000.00

2. Actividad No 02 – Reconociendo la estructura

1. Se requiere determinar la estructura de la tabla DEPARTMENTS y sus datos.

SP_HELP 'DEPARTMENTS'

	Name	Owner	Type	Created_datetime
1	departments	dbo	user table	2018-10-03 09:42:23.247

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim TrailingBlanks	FixedLenNullInSource
1	department_id	int	no	4	10	0	no	(n/a)	(n/a)
2	department_name	varchar	no	30			yes	no	yes
3	manager_id	int	no	4	10	0	yes	(n/a)	(n/a)
4	location_id	int	no	4	10	0	yes	(n/a)	(n/a)

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data located on filegroup
1	PRIMARY

	index_name	index_description	index_keys
1	dept_id_pk	clustered, unique, primary key located on PRIMARY	department_id

2. El departamento de Recursos Humanos requiere un reporte que muestre los campos: employee_id, last_name y job_id, asicomo el campo hire_date con el alias StartDate.

```
SELECT emp.employee_id,
emp.last_name,
emp.job_id,
emp.hire_date AS StartDate
FROM employees AS emp;
```

	employee_id	last_name	job_id	StartDate
1	100	King	AD_PRES	2003-06-17
2	101	Kochhar	AD_VP	2005-09-21
3	102	De Haan	AD_VP	2001-01-13
4	103	Hunold	IT_PROG	2006-01-03
5	104	Ernst	IT_PROG	2007-05-21
6	105	Austin	IT_PROG	2005-06-25
7	106	Pataballa	IT_PROG	2006-02-05
8	107	Lorentz	IT_PROG	2007-02-07
9	108	Greenberg	FI_MGR	2002-08-17
10	109	Faviet	FI_ACCOUNT	2002-08-16
11	110	Chen	FI_ACCOUNT	2005-09-28

3. Finalmente el departamento de Recursos Humanos requiere un listado de todos valores del campo JOB.ID de la tabla EMPLOYEES pero que se muestren de forma única y no repetida.

```
SELECT DISTINCT job_id FROM employees;
```

	job_id
1	AC_ACCOUNT
2	AC_MGR
3	AD_ASST
4	AD PRES
5	AD_VP
6	FI_ACCOUNT
7	FI_MGR
8	HR_REP
9	IT_PROG
10	MK_MAN
11	MK_REP

3. Actividad No 03 – Consultas Básicas

1. El departamento de Recursos Humanos requiere ampliar el reporte anterior (4.2.2) para hacerlo más comprensible, por lo que se requiere que los encabezados de las columnas sean: Emp No, Empleado, Puesto y Fecha Contratación.

```
SELECT emp.employee_id AS 'Emp N',  
emp.last_name AS Empleado,  
emp.job_id AS Puesto,  
emp.hire_date AS 'Fecha de contratación'  
FROM employees AS emp;
```

	Emp N	Empleado	Puesto	Fecha de contratacion
1	100	King	AD_PRES	2003-06-17
2	101	Kochhar	AD_VP	2005-09-21
3	102	De Haan	AD_VP	2001-01-13
4	103	Hunold	IT_PROG	2006-01-03
5	104	Ernst	IT_PROG	2007-05-21
6	105	Austin	IT_PROG	2005-06-25
7	106	Pataballa	IT_PROG	2006-02-05
8	107	Lorentz	IT_PROG	2007-02-07
9	108	Greenb...	FI_MGR	2002-08-17
10	109	Faviet	FI_ACC...	2002-08-16
11	110	Chen	FI_ACC...	2005-09-28
12	111	Sciarra	FI_ACC...	2005-09-30
13	112	Uman	FI_ACC...	2006-03-07
14	113	Popp	FI_ACC...	2007-12-07
15	114	Raphaely	PU_MAN	2002-12-07
16	115	Khoo	PU_CLE...	2003-05-18
17	116	Beiko	PU_CLE...	2005-12-24

2. Adicionalmente el departamento de Recursos Humanos requiere un reporte más sencillo, en el que se muestre los campos: last_name y job_id en una sola y única columna (los datos deben estar separados por una coma) que tenga como alias Empleado y Puesto.

```
SELECT CONCAT(emp.last_name,',',emp.job_id) AS 'Empleado y Puesto'  
  
FROM employees AS emp;
```

	Empleado y Puesto
1	King,AD_PRES
2	Kochhar,AD_VP
3	De Haan,AD_VP
4	Hunold,IT_PROG
5	Ernst,IT_PROG
6	Austin,IT_PROG
7	Pataballa,IT_PR...
8	Lorentz,IT_PROG
9	Greenberg,FI_MGR
10	Faviet,FI_ACCOU...
11	Chen,FI_ACCOU...
12	Sciarra,FI_ACCO...
13	Urman,FI_ACCO...
14	Popp,FI_ACCOU...
15	Raphaely,PU_MAN
16	Khoo,PU_CLERK
17	Baida,PU_CLERK
18	Tobias,PU_CLERK

3. Finalmente a modo de práctica, realizar una consulta que muestre todos los campos de la tabla EMPLOYEES, en una sola y única columna, los datos deben estar separados por una coma y la columna debe tener como encabezado Los Empleados

```
SELECT CONCAT(emp.employee_id,',',
emp.first_name,',',
emp.last_name,',',
emp.email,',',
emp.phone_number,',',
emp.hire_date,',',
emp.job_id,',',
emp.salary,',',
emp.commission_pct,',',
emp.manager_id,',',
emp.department_id) AS 'Los empleados'
FROM employees AS emp;
```


empleados	
1	100.Steven.King,SKING,515.123.4567,2003-06-17,AD_P...
2	101.Neena.Kochhar,NKOCHHAR,515.123.4568,2005-09-...
3	102.Lex.De Haan,LDEHAAN,515.123.4569,2001-01-13,A...
4	103.Alexander.Hunold,AHUNOLD,590.423.4567,2006-01-...
5	104.Bruce.Ernst,BERNST,590.423.4568,2007-05-21,IT_P...
6	105.David.Austin,DAUSTIN,590.423.4569,2005-06-25,IT_...
7	106.Valli.Pataballa,VPATABAL,590.423.4560,2006-02-05,I...
8	107.Diana.Lorentz,DLORENTZ,590.423.5567,2007-02-07,...
9	108.Nancy.Greenberg,NGREENBE,515.124.4569,2002-08-...
10	109.Daniel.Fayiet,DFAVIET,515.124.4169,2002-08-16,FI_...
11	110.John.Chen,JCHEN,515.124.4269,2005-09-28,FI_ACC...
12	111.Ismael.Sciara,ISCIARRA,515.124.4369,2005-09-30,FI...
13	112.Jose Manuel Uman,JMURMAN,515.124.4469,2006-0...
14	113.Luis.Popp,LPOPP,515.124.4567,2007-12-07,FI_ACCO...
15	114.Den.Raphaely,DRAPHEAL,515.127.4561,2002-12-07,...

4. Actividad No 04 – Restricción y Ordenamiento

1. Debido a problemas con el presupuesto, el departamento de Recursos Humanos requiere un reporte que muestre los apellidos (last_name) y salarios (salary) de todos los empleados que ganen más de \$ 12,000.

```
select last_name,salary from employees where salary >12000;
```

	last_name	salary
1	King	24000.00
2	Kochhar	17000.00
3	De Haan	17000.00
4	Greenberg	12008.00
5	Russell	14000.00
6	Partners	13500.00
7	Hartstein	13000.00
8	Higgins	12008.00

2. Asimismo se requiere realizar una consulta que muestre los apellidos (last_name) y el número de departamento (department_id) para los empleados que tengan numero (employee_id) 176.

```
select last_name,department_id from employees where employee_id =176;
```

	last_name	department_id	employee_id
1	King	90	176
2	Kochhar	90	176
3	De Haan	90	176
4	Hunold	60	176
5	Ernst	60	176
6	Austin	60	176
7	Pataballa	60	176
8	Lorentz	60	176
9	Greenberg	100	176
10	Faviet	100	176
11	Chen	100	176
12	Sciarra	100	176
13	Uman	100	176
14	Popp	100	176

3. El departamento de Recursos Humanos necesita determinar los mayores y menores sueldos, modificar la consulta del ítem 4.1. para mostrar el apellido y salario de cada empleado cuyo sueldo no esté en el rango de \$ 5,000 a \$ 12,000.

```
select last_name,job_id,salary as Sal from employees where salary <5000 and salary <12000;
```

	last_name	job_id	Sal
1	Hunold	IT_PROG	9000.00
2	Ernst	IT_PROG	6000.00
3	Faviet	FI_ACCOUNT	9000.00
4	Chen	FI_ACCOUNT	8200.00
5	Sciarra	FI_ACCOUNT	7700.00
6	Urman	FI_ACCOUNT	7800.00
7	Popp	FI_ACCOUNT	6900.00
8	Raphaely	PU_MAN	11000...
9	Weiss	ST_MAN	8000.00
10	Fripp	ST_MAN	8200.00
11	Kaufling	ST_MAN	7900.00
12	Vollman	ST_MAN	6500.00
13	Mourgos	ST_MAN	5800.00
14	Cambrault	SA_MAN	11000...
15	Zlotkey	SA_MAN	10500...

4. Crear un reporte que muestre los apellidos (last_name), puesto (job_id) y fecha de contratación (hire_date), de los empleados que apellidan 'Matos' y 'Taylor', asimismo presentar el reporte ordenado ascendentemente por fecha de contratación.

```
select last_name,job_id,hire_date from employees where last_name = 'Matos' or last_name = 'Taylor' order by hire_date asc;
```

	last_name	job_id	hire_date
1	Taylor	SH_CLERK	2006-01-24
2	Matos	ST_CLERK	2006-03-15
3	Taylor	SA_REP	2006-03-24

5. Mostrar los apellidos (last_name) y número de departamento (departamento_id) de todos los empleados que pertenezcan a los departamentos 20 o 50 en orden alfabético ascendente por el apellido.

```
select last_name,department_id from employees where department_id = 20 or department_id = 50 order by last_name asc;
```


select last_name,job_id from employees where manager_id is null;

	last_name	job_id
1	King	AD_PRES

9. Crear un reporte para mostrar los apellidos (last_name), salario (salary) y % de comisión (commission_pct). Ordenar los datos por salario y comisión de manera descendente, utilizar la opción numérica de la cláusula ORDER BY.

select last_name,salary,commission_pct from employees order by salary desc,commission_pct desc;

	last_name	salary	commission_pct
1	King	24000.00	NULL
2	Kochhar	17000.00	NULL
3	De Haan	17000.00	NULL
4	Russell	14000.00	0.40
5	Partners	13500.00	0.30
6	Hartstein	13000.00	NULL
7	Higgins	12008.00	NULL
8	Greenberg	12008.00	NULL
9	Erazuriz	12000.00	0.30
10	Ozer	11500.00	0.25
11	Cambrault	11000.00	0.30

10. El personal del departamento de Recursos Humanos desea tener mayor flexibilidad con los reportes hechos. Por ejemplo se requiere un reporte de los apellidos (last_name) y salarios (salary) de todos los empleados que tengan un salario mayor a un monto que el personal de Recursos Humanos ingresará. Probar con el valor \$ 12,000.

declare @salario as decimal(9,2); set @salario = 12000; select last_name,salary from employees where salary > @salario;

	last_name	salary
1	King	24000.00
2	Kochhar	17000.00
3	De Haan	17000.00
4	Greenberg	12008.00
5	Russell	14000.00
6	Partners	13500.00
7	Hartstein	13000.00
8	Higgins	12008.00

11. El departamento de Recursos Humanos requiere extraer reporte basados en el Administrador (manager_id). Se requiere crear una consulta que pregunte al usuario por el Administrador (manager_id) y genere un reporte con los números de empleado (employee_id), apellidos (last_name),

salarios (salary) y numero de departamento de los empleados que este Administrador tiene a su cargo. Adicionalmente también se desea tener la habilidad de ordenar este reporte en base a una determinada columna. Probar con los siguientes valores:

Administrador (manager_id) = 103, ordenado por Apellido (last_name)

Administrador (manager_id) = 201, ordenado por Salario (salary)

Administrador (manager_id) = 124, ordenado por No de Empleado (employee_id)

```
declare @gerente as int;
set @gerente = 103;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by last_name;
set @gerente = 201;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by salary;
set @gerente = 124;
select employee_id,last_name,salary,department_id from employees where manager_id = @gerente order by employee_id;
go
```

	employee_id	last_name	salary	department_id
1	105	Austin	4800.00	60
2	104	Ernst	6000.00	60
3	107	Lorentz	4200.00	60
4	106	Pataballa	4800.00	60

	employee_id	last_name	salary	department_id
1	202	Fay	6000.00	20

	employee_id	last_name	salary	department_id
1	141	Rajs	3500.00	50
2	142	Davies	3100.00	50
3	143	Matos	2600.00	50
4	144	Vargas	2500.00	50
5	196	Walsh	3100.00	50
6	197	Feeney	3000.00	50
7	198	OConnell	2600.00	50
8	199	Grant	2600.00	50

- Generar un listado de apellidos (last_name) de todos los empleados que tengan la letra 'a' en la tercera letra de su apellido.

```
select last_name from employees where SUBSTRING(last_name,3,1) = 'a';
go
```

	last_name
1	Grant
2	Grant
3	Whalen

13. Mostrar los apellidos (last_name) de todos los empleados que tengan tanto la letra 'a' como la letra 'e' en su apellido.

```
select last_name from employees where SUBSTRING(last_name,3,1) = 'a' or SUBSTRING(last_name,3,1) = 'e';  
go
```

	last_name
1	Greenberg
2	Chen
3	Gee
4	McEwen
5	Greene
6	Lee
7	Ozer
8	Abel
9	Grant
10	Fleaur
11	Everett
12	Feeney
13	Grant
14	Whalen
15	Baer
16	Gietz

14. Mostrar los apellidos (last_name), puestos (job_id) y salario (salary) de todos los empleados que sean Representantes de Ventas (SA_REP) o Responsables de Inventario (ST_CLERK) y cuyos salarios no sean iguales a \$ 2,500, \$ 3,500 o \$ 7,000.

```
select last_name,job_id,salary from employees where (job_id = 'SA_REP' or job_id = 'ST_CLERK')  
and (salary = 2500 or salary = 3500 or salary = 7000);  
go
```

	last_name	job_id	salary
1	Marlow	ST_CLERK	2500.00
2	Patel	ST_CLERK	2500.00
3	Rajs	ST_CLERK	3500.00
4	Vargas	ST_CLERK	2500.00
5	Tuvault	SA_REP	7000.00
6	Sewall	SA_REP	7000.00
7	Grant	SA_REP	7000.00

15. Modificar el reporte del ítem 4.6 y mostrar adicionalmente los datos de comisión (commission_pct) de todos los empleados que solamente el 20 % de comisión.

```
select last_name 'Empleado',salary 'Salario Mensual',commission_pct from employees where salary >5000 and salary <12000 and (department_id = 20 or department_id = 50) and commission_pct = 0.20;
go
```

Empleado	Salario Mensual	commission_pct

5. Actividad No 05 – Funciones

1. Se requiere realizar una consulta que visualice la fecha del sistema.

```
SELECT CONVERT (date, SYSDATETIME())
,CONVERT (date, SYSDATETIMEOFFSET())
,CONVERT (date, SYSUTCDATETIME())
,CONVERT (date, CURRENT_TIMESTAMP)
,CONVERT (date, GETDATE())
,CONVERT (date, GETUTCDATE());
```

	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)
1	2018/05/01	2018/05/01	2018/05/01	2018/05/01	2018/05/01

2. El departamento de Recursos Humanos necesita un reporte de todos los empleados que muestre el No de Empleado, Apellidos, Salario y una columna más con el cálculo del salario incrementado en 15.5 % (expresado solo en enteros) esta columna debe etiquetarse Nuevo Salario

```
SELECT employee_id,last_name,salary,salary*0.155 as newsalary FROM employees
```

	employee_id	last_name	salary	newsalary
1	100	King	24000.00	3720.00000
2	101	Kochhar	17000.00	2635.00000
3	102	De Haan	17000.00	2635.00000
4	103	Hunold	9000.00	1395.00000
5	104	Ernst	6000.00	930.00000
6	105	Austin	4800.00	744.00000
7	106	Pataballa	4800.00	744.00000
8	107	Lorentz	4200.00	651.00000
9	108	Greenb...	12008.00	1861.24000
10	109	Faviet	9000.00	1395.00000
11	110	Chen	8200.00	1271.00000
12	111	Sciarra	7700.00	1193.50000
13	112	Uman	7800.00	1209.00000
14	113	Popp	6900.00	1069.50000
15	114	Raphaely	11000.00	1705.00000
16	115	Khoo	3100.00	480.50000

3. Modificar la consulta anterior y adicionar una columna que muestre el resultado de la resta entre el antiguo salario y el nuevo salario. Etiquetar esta columna como Incremento.

```
SELECT employee_id,last_name,salary,salary*0.155 as newsalary,salary-(salary*0.155) as incremento FROM employees
```

	employee_id	last_name	salary	newsalary	incremento
1	100	King	24000.00	3720.00000	20280.00000
2	101	Kochhar	17000.00	2635.00000	14365.00000
3	102	De Haan	17000.00	2635.00000	14365.00000
4	103	Hunold	9000.00	1395.00000	7605.00000
5	104	Ernst	6000.00	930.00000	5070.00000
6	105	Austin	4800.00	744.00000	4056.00000
7	106	Pataballa	4800.00	744.00000	4056.00000
8	107	Lorentz	4200.00	651.00000	3549.00000
9	108	Greenb...	12008.00	1861.24000	10146.76000
10	109	Faviet	9000.00	1395.00000	7605.00000
11	110	Chen	8200.00	1271.00000	6929.00000

4. Crear un reporte que muestre los Apellidos (con la primera letra en Mayúsculas y las demás en Minúsculas) y la longitud de los apellidos (colocar alias Longitud), para todos aquellos empleados quienes sus apellidos empiecen con las letras 'J', 'A' y 'M'. Ordenar los resultados por la columna Apellido.

```

select UPPER(last_name) "Apellido", (LOWER(first_name)) "Longitud"
from employees
where last_name like 'A %'
or last_name like 'J %'
or last_name like 'M %' order by last_name asc;

```

	Apellido	Longitud
1	ABEL	ellen
2	ANDE	sundar
3	ATKINSON	mozhe
4	AUSTIN	david
5	JOHNSON	charles
6	JONES	vance
7	MALLIN	jason
8	MARKLE	steven
9	MARLOW	james
10	MARVINS	mattea
11	MATOS	randall

5. Modificar la consulta anterior a fin de que consulte primero al usuario con que letra empieza el apellido a buscar. Considerar que no importa si la letra esta mayúscula o minúscula de igual manera debe mostrar los resultados.

```

select initcap(FIRST_NAME) as "name", length(first_name) as "Length" from employees where
upper(substr(first_name,1,1))=upper('&Inicial') order by first_name;

```

6. El departamento de Recursos Humanos la duración o tiempo de permanencia de cada empleado, mostrar el Apellido y el calculo del número de meses entre la fecha de hoy y la fecha en que fue contratado el empleado, Etiquetar la columna como Meses Trabajados, ordenar los resultados por el resultado de los números de meses, Redondear el número de meses al entero más cercano.

```

SELECT LAST_NAME, ROUND(MONTHS_BETWEEN(SYSDATE,HIRE_DATE),0) "MONTHS_WORK"
from employees order by MONTHS_BETWEEN( HIRE_DATE, SYSDATE);

```

7. Crear una consulta que devuelva los Apellidos y Salarios de todos los empleados, Formatear la columna salario para que muestre 15 caracteres, completar con el símbolo '\$' los espacios previos al valor de la columna salario, ejemplo: \$\$\$\$\$\$\$\$\$\$10000. Etiquetar esta columna como Salario.

```

CREATE FUNCTION LPAD
(
  @string VARCHAR(MAX),
  @length INT,
  @pad CHAR
)

```

```

RETURNS VARCHAR(MAX)
AS
BEGIN
RETURN REPLICATE(@pad, @length - LEN(@string)) + @string;
END
GO
SELECT dbo.LPAD(salary, 15, '$') VALUE
FROM employees;

```

	value
1	\$\$\$\$\$\$\$24000.00
2	\$\$\$\$\$\$\$17000.00
3	\$\$\$\$\$\$\$17000.00
4	\$\$\$\$\$\$\$9000.00
5	\$\$\$\$\$\$\$6000.00
6	\$\$\$\$\$\$\$4800.00
7	\$\$\$\$\$\$\$4800.00
8	\$\$\$\$\$\$\$4200.00
9	\$\$\$\$\$\$\$12008.00
10	\$\$\$\$\$\$\$9000.00
11	\$\$\$\$\$\$\$8200.00

8. Crear una consulta que muestre en una única columna los primeros 8 caracteres del apellido de los empleados e indique sus salarios representados por asteriscos (*), cada asterisco representa el valor 1000. Ordenar el listado por el salario de los empleados. Asimismo Etiquetar la columna como 'Empleados y sus Salarios'.
9. Finalmente crear una consulta que muestre los Apellidos de los empleados y el No de Semanas Empleado hasta la actualidad para todos los empleados del departamento No 90, truncar el número de semanas a sin decimales. Ordenar el resultado por el No de Semanas y etiquetar la columna como tenencia.

```

select last_name, TRUNC(((SYSDATE-hire_date)/7),0) as TENURE from employees where department_id=90 ORDER BY hire_date DESC;

```

6. Actividad No 06 – Funciones de Conversión

1. Crear un reporte que muestre lo siguiente por cada empleado.
(Apellido del empleado) gana (Salario) pero quisiera (3 veces Salario).
Etiquetar la columna como Sueldos Soñados.

```
select 'Sueldos Soñados'=(last_name + ' gana ' + Cast(salary as varchar(18)) + ' pero  
quisiera ' + Cast((salary * 3) as varchar(18)))  
from dbo.employees  
go
```



	sueldos soñados
1	Kinggana: 24000.00pero quisiera: 72000.00
2	Kochhargana: 17000.00pero quisiera: 51000.00
3	De Haangana: 17000.00pero quisiera: 51000.00
4	Hunoldgana: 9000.00pero quisiera: 27000.00
5	Ernstgana: 6000.00pero quisiera: 18000.00
6	Austingana: 4800.00pero quisiera: 14400.00
7	Pataballagana: 4800.00pero quisiera: 14400.00
8	Lorentzgana: 4200.00pero quisiera: 12600.00
9	Greenberggana: 12008.00pero quisiera: 36024.00
10	Faviatgana: 9000.00pero quisiera: 27000.00
11	Chengana: 8200.00pero quisiera: 24600.00
12	Sciaragana: 7700.00pero quisiera: 23100.00
13	Umangana: 7800.00pero quisiera: 23400.00
14	Poppgana: 6900.00pero quisiera: 20700.00
15	Raphaelygana: 11000.00pero quisiera: 33000.00
16	Khoogana: 3100.00pero quisiera: 9300.00
17	Baidagana: 2900.00pero quisiera: 8700.00
18	Tobiasgana: 2800.00pero quisiera: 8400.00
19	Himurogana: 2600.00pero quisiera: 7800.00
20	Colmenaresgana: 2500.00pero quisiera: 7500.00
21	Weissgana: 8000.00pero quisiera: 24000.00
22	Frippgana: 8200.00pero quisiera: 24600.00
23	Kauflinggana: 7900.00pero quisiera: 23700.00
24	Vollmangana: 6500.00pero quisiera: 19500.00
25	Mourgosgana: 5800.00pero quisiera: 17400.00
26	Nayergana: 3200.00pero quisiera: 9600.00

2. Realizar una consulta que muestre el Apellido del empleado, fecha de contratación y la Fecha de Revisión del Salario, la cual es el primer Lunes después de cada seis meses de servicio, etiquetar la columna como Revisión, asimismo el formato de esta fecha debe ser similar al siguiente:

Lunes, el veintiuno de julio, 2003

```
select last_name, hire_date as Revision from employees  
where hire_date between '2003-06-17' and '2005-09-21';  
go
```

98 % 0 cambios 0 autores, 0 cambios

T-SQL Resultados Mensaje

	last_name	Revision
1	King	2003-06-17
2	Kochhar	2005-09-21
3	Austin	2005-06-25
4	Tobias	2005-07-24
5	Weiss	2004-07-18
6	Fripp	2005-04-10
7	Nayer	2005-07-16
8	Bissot	2005-08-20
9	Marlow	2005-02-16
10	Mallin	2004-06-14

ocal) (13.0 RTM) | DEPI22\DEPI18 (58) | Human_Resources | 00:00:00 | 35 filas

3. Mostrar un reporte que tenga los Apellidos, Fecha de Contratación y el Día de Inicio de cada empleado (Lunes, Martes, etc...), etiquetar la última columna como Día. Ordenar los resultados por el Día de Inicio empezando por Lunes.

```
select e.last_name, e.hire_date, DateName(WEEKDAY, jh.START_DATE)as 'Dia'
from dbo.employees as e inner join dbo.job_history as jh on
e.employee_id=jh.employee_id
go
```

98 % 0 cambios 0 autores, 0 cambios

T-SQL Resultados Mensaje

	last_name	hire_date	Dia
1	Kochhar	2005-09-21	Domingo
2	Kochhar	2005-09-21	Domingo
3	De Haan	2001-01-13	Sábado
4	Raphaely	2002-12-07	Viernes
5	Kaufling	2003-05-01	Lunes
6	Taylor	2006-03-24	Viernes
7	Taylor	2006-03-24	Lunes
8	Whalen	2003-09-17	Domingo
9	Whalen	2003-09-17	Lunes
10	Hartstein	2004-02-17	Martes

ocal) (13.0 RTM) | DEPI22\DEPI18 (60) | Human_Resources | 00:00:00 | 10 filas

4. Crear un listado que muestre los Apellidos de los empleados y sus Montos de Comisión, en caso no tenga comisión deberá mostrar el texto 'Sin Comisión', etiquetar esta ultima columna como Comisión.

```
select last_name as 'Apellidos', 'Comision'='Sin Comision' from dbo.employees where
commission_pct = 0
UNION
select last_name as 'Apellidos', 'Comision'= Cast((salary * commission_pct) as
varchar(20)) from dbo.employees where commission_pct >0
go
```

	Apellidos	Comision
1	Abel	3300.0000
2	Ande	640.0000
3	Banda	620.0000
4	Bates	1095.0000
5	Bernstein	2375.0000
6	Bloom	2000.0000
7	Cambr...	1500.0000
8	Cambr...	3300.0000
9	Doran	2250.0000
10	Erazuriz	3600.0000

ocal) (13.0 RTM) | DEPI22\DEPI18 (56) | Human_Resources | 00:00:00 | 35 filas

5. Utilizando la función DECODE, crear un reporte que muestre los apellidos, los puestos y los grados de los empleados basados en sus puestos, utilizando la siguiente información:

Puesto	Grado
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
Ninguno de los Anteriores	0

6. Rescribir la consulta anterior utilizando la función CASE.

```

select e.last_name as 'Apellidos', j.job_title, case
when j.job_id = 'AD_PRES' THEN 'A'
when j.job_id = 'ST_MAN' THEN 'B'
when j.job_id = 'IT_PROG' THEN 'C'
when j.job_id = 'SA_REP' THEN 'D'
else '0' END as 'Grados' from dbo.employees as e inner join dbo.jobs as j on
e.job_id=j.job_id
go

```

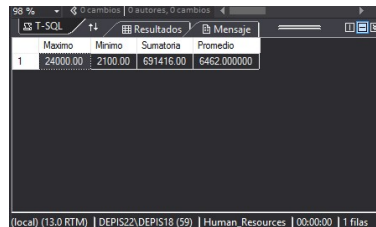
	Apellidos	job_title	Grados
1	King	President	A
2	Kochhar	Administration Vice President	0
3	De Haan	Administration Vice President	0
4	Hunold	Programmer	C
5	Ernst	Programmer	C
6	Austin	Programmer	C
7	Pataballa	Programmer	C
8	Lorentz	Programmer	C
9	Greenb...	Finance Manager	0
10	Faviet	Accountant	0

cal) (13.0 RTM) | DEPI22\DEPI18 (57) | Human_Resources | 00:00:00 | 107 filas

7. Actividad No 07 – Funciones de Agrupación

1. El departamento de Recursos Humanos requiere un reporte que muestre el máximo, el mínimo, la suma y el promedio de los salarios de todos los empleados, Etiquetar esta columnas como Máximo, Mínimo, Suma y Promedio respectivamente, Redondear estos valores a enteros sin decimales.

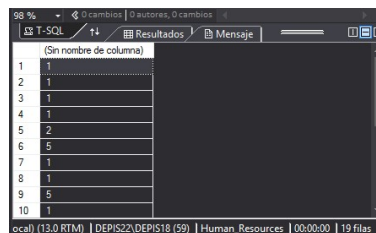
```
SELECT ROUND(MAX(salary),0) AS "Maximo", ROUND(MIN(salary),0) AS "Minimo", ROUND(SUM(salary),0) AS "Sumatoria", ROUND(AVG(salary),0) AS "Promedio"
FROM employees;
```



	Maximo	Minimo	Sumatoria	Promedio
1	24000.00	2100.00	691416.00	6462.000000

2. Modificar la consulta anterior para mostrar el máximo, mínimo, suma y promedio de los salarios por cada Puesto de trabajo.
3. Realizar un reporte que muestre la cantidad de empleados por Puesto de trabajo. Con la opción de que el usuario pueda ingresar todos los puestos o uno solo.

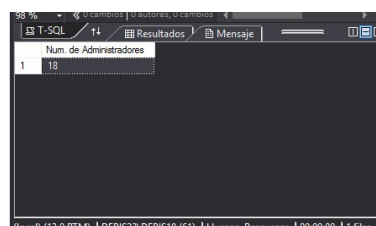
```
SELECT COUNT(*)
FROM employees
GROUP BY job_id;
```



job_id	count(*)
1	1
2	1
3	1
4	1
5	2
6	5
7	1
8	1
9	5
10	1

4. Determinar el número de Administradores o Supervisores utilizar la columna manager_id para esto. Etiquetar la columna como No de Administradores

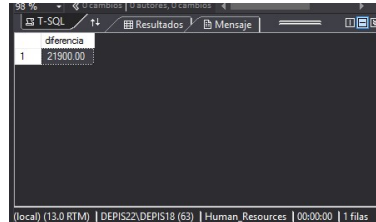
```
SELECT COUNT(DISTINCT manager_id) AS "Numero de Administradores"
FROM employees;
```



manager_id	count(DISTINCT manager_id)
1	18

5. Encontrar la diferencia entre el máximo y mínimo salario de los empleados. Etiquetar la columna como Diferencia

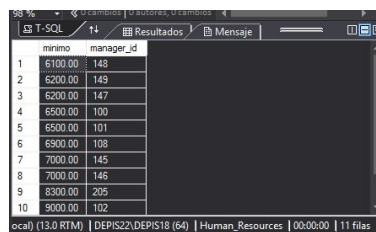
```
SELECT (MAX(salary) - MIN(salary)) AS "diferencia"
FROM employees;
```



diferencia
21900.00

6. Crear un reporte que muestre los No de Administradores (manager_id) y el salario de su empleado peor pagado. Excluir a los empleados cuyo Administrador no se conozca. Excluir asimismo cualquier grupo cuyo salario mínimo sea \$6000 o menos. Ordenar los resultados por el mínimo salario en forma descendente.

```
SELECT salman.minimo,
salman.manager_id
FROM (SELECT MIN(salary) AS 'minimo',
manager_id
FROM employees
WHERE salary > 6000
GROUP BY manager_id) AS salman
ORDER BY salman.minimo;
```



	minimo	manager_id
1	6100.00	148
2	6200.00	149
3	6200.00	147
4	6500.00	100
5	6500.00	101
6	6500.00	108
7	7000.00	145
8	7000.00	146
9	8300.00	205
10	9000.00	102

7. Crear una consulta que muestre el número total de empleados, así como el número total de empleados contratados en los años 1995, 1996, 1997 y 1998, etiquetar las columnas apropiadamente.
8. Crear una consulta matriz que muestre el puesto, el salario por cada puesto basado en el No de Departamento del empleado y el total del salario para cada puesto para los departamento 20, 50, 80 y 90, colocar un nombre apropiado a cada columna.

8. Actividad No 08 – Enlaces

1. El departamento de Recursos Humanos requiere un reporte que muestre las direcciones de todos los departamentos. Utilizar las tablas LOCATIONS y COUNTRIES. Mostrar el ID de la Ubicación (location_id), dirección (street_address), ciudad (city), estado o provincia (state_province) y país (country_name).

```
select l.location_id , l.street_address , l.city , l.state_province , c.country_name
from locations as l
join countries as c
on l.country_id = c.country_id
```

Results	Messages
1	1000 1287 Nieuw Zeeland
2	1000 1288 New Zealand
3	1000 1289 New Zealand
4	1000 1290 New Zealand
5	1000 1291 New Zealand
6	1000 1292 New Zealand
7	1000 1293 New Zealand
8	1000 1294 New Zealand
9	1000 1295 New Zealand
10	1000 1296 New Zealand
11	1000 1297 New Zealand
12	1000 1298 New Zealand
13	1000 1299 New Zealand
14	1000 1300 New Zealand
15	1000 1301 New Zealand
16	1000 1302 New Zealand
17	1000 1303 New Zealand
18	1000 1304 New Zealand
19	1000 1305 New Zealand
20	1000 1306 New Zealand
21	1000 1307 New Zealand
22	1000 1308 New Zealand
23	1000 1309 New Zealand
24	1000 1310 New Zealand
25	1000 1311 New Zealand
26	1000 1312 New Zealand
27	1000 1313 New Zealand
28	1000 1314 New Zealand
29	1000 1315 New Zealand
30	1000 1316 New Zealand
31	1000 1317 New Zealand
32	1000 1318 New Zealand
33	1000 1319 New Zealand
34	1000 1320 New Zealand
35	1000 1321 New Zealand
36	1000 1322 New Zealand
37	1000 1323 New Zealand
38	1000 1324 New Zealand
39	1000 1325 New Zealand
40	1000 1326 New Zealand
41	1000 1327 New Zealand
42	1000 1328 New Zealand
43	1000 1329 New Zealand
44	1000 1330 New Zealand
45	1000 1331 New Zealand
46	1000 1332 New Zealand
47	1000 1333 New Zealand
48	1000 1334 New Zealand
49	1000 1335 New Zealand
50	1000 1336 New Zealand
51	1000 1337 New Zealand
52	1000 1338 New Zealand
53	1000 1339 New Zealand
54	1000 1340 New Zealand
55	1000 1341 New Zealand
56	1000 1342 New Zealand
57	1000 1343 New Zealand
58	1000 1344 New Zealand
59	1000 1345 New Zealand
60	1000 1346 New Zealand
61	1000 1347 New Zealand
62	1000 1348 New Zealand
63	1000 1349 New Zealand
64	1000 1350 New Zealand
65	1000 1351 New Zealand
66	1000 1352 New Zealand
67	1000 1353 New Zealand
68	1000 1354 New Zealand
69	1000 1355 New Zealand
70	1000 1356 New Zealand
71	1000 1357 New Zealand
72	1000 1358 New Zealand
73	1000 1359 New Zealand
74	1000 1360 New Zealand
75	1000 1361 New Zealand
76	1000 1362 New Zealand
77	1000 1363 New Zealand
78	1000 1364 New Zealand
79	1000 1365 New Zealand
80	1000 1366 New Zealand
81	1000 1367 New Zealand
82	1000 1368 New Zealand
83	1000 1369 New Zealand
84	1000 1370 New Zealand
85	1000 1371 New Zealand
86	1000 1372 New Zealand
87	1000 1373 New Zealand
88	1000 1374 New Zealand
89	1000 1375 New Zealand
90	1000 1376 New Zealand
91	1000 1377 New Zealand
92	1000 1378 New Zealand
93	1000 1379 New Zealand
94	1000 1380 New Zealand
95	1000 1381 New Zealand
96	1000 1382 New Zealand
97	1000 1383 New Zealand
98	1000 1384 New Zealand
99	1000 1385 New Zealand
100	1000 1386 New Zealand
101	1000 1387 New Zealand
102	1000 1388 New Zealand
103	1000 1389 New Zealand
104	1000 1390 New Zealand
105	1000 1391 New Zealand
106	1000 1392 New Zealand
107	1000 1393 New Zealand
108	1000 1394 New Zealand
109	1000 1395 New Zealand
110	1000 1396 New Zealand
111	1000 1397 New Zealand
112	1000 1398 New Zealand
113	1000 1399 New Zealand
114	1000 1400 New Zealand
115	1000 1401 New Zealand
116	1000 1402 New Zealand
117	1000 1403 New Zealand
118	1000 1404 New Zealand
119	1000 1405 New Zealand
120	1000 1406 New Zealand
121	1000 1407 New Zealand
122	1000 1408 New Zealand
123	1000 1409 New Zealand
124	1000 1410 New Zealand
125	1000 1411 New Zealand
126	1000 1412 New Zealand
127	1000 1413 New Zealand
128	1000 1414 New Zealand
129	1000 1415 New Zealand
130	1000 1416 New Zealand
131	1000 1417 New Zealand
132	1000 1418 New Zealand
133	1000 1419 New Zealand
134	1000 1420 New Zealand
135	1000 1421 New Zealand
136	1000 1422 New Zealand
137	1000 1423 New Zealand
138	1000 1424 New Zealand
139	1000 1425 New Zealand

2. El departamento de Recursos Humanos necesita un reporte de todos empleados, que muestres los apellidos de empleado (`last_name`), el No de departamento (`department_id`) y el nombre del departamento (`department_name`) al cual pertenece.

```
select e.last_name , d.department_id , d.department_name from employees as e
left join departments as d
on e.department_id = d.department_id order by d.department_name;
```

[illegible]

3. El departamento de Recursos Humanos necesita un reporte de los empleados de la ciudad de Toronto. Mostrar los Apellidos, Puesto, No de Departamento y Nombre de Departamento de todos los empleados que trabajan en Toronto.

```
select e.last_name , e.department_id, j.job_title, d.department_name , l.city
from employees as e
left join jobs as j
on e.job_id = j.job_id
join departments as d
on e.department_id=d.department_id
```

join locations as l
on d.location_id = l.location_id
where l.city='Toronto';

Query: SELECT e.last_name, e.department_id, d.job_title, d.department_name, l.city FROM employees e, departments d, locations l WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND l.city = 'Toronto';

Row	last_name	department_id	job_title	department_name	city
1	De Haan	90	Managing Manager	Marketing	Toronto
2	Papp	90	Managing Human Resources	Marketing	Toronto

4. Crear un reporte que muestre los Apellidos y No de Identificación de los empleados, asimismo también debe mostrarse el Apellido y No de Identificación de su Administrador.

SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
join employees m
ON (e.manager_id = m.employee_id)

Query: SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado', m.employee_id 'ID_Manager', m.last_name 'Manager' FROM employees e JOIN employees m ON (e.manager_id = m.employee_id);

Row	ID_Empleado	Empleado	ID_Manager	Manager
1	100	De Haan	100	King
2	101	De Haan	100	King
3	102	De Haan	100	King
4	103	De Haan	100	King
5	104	De Haan	100	King
6	105	De Haan	100	King
7	106	De Haan	100	King
8	107	De Haan	100	King
9	108	De Haan	100	King
10	109	De Haan	100	King
11	110	De Haan	100	King
12	111	De Haan	100	King
13	112	De Haan	100	King
14	113	De Haan	100	King
15	114	De Haan	100	King
16	115	De Haan	100	King
17	116	De Haan	100	King
18	117	De Haan	100	King
19	118	De Haan	100	King
20	119	De Haan	100	King

5. Modificar la consulta anterior para que incluya también a los empleados quienes no tienen Administrador asignado.

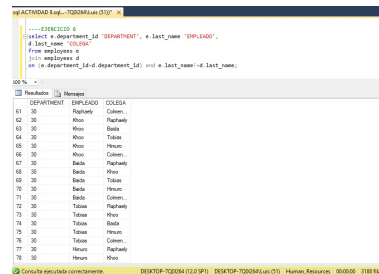
SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado',
m.employee_id 'ID_Manager', m.last_name 'Manager'
FROM employees e
left outer join employees m
ON (e.manager_id = m.employee_id)

Query: SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado', m.employee_id 'ID_Manager', m.last_name 'Manager' FROM employees e LEFT OUTER JOIN employees m ON (e.manager_id = m.employee_id);

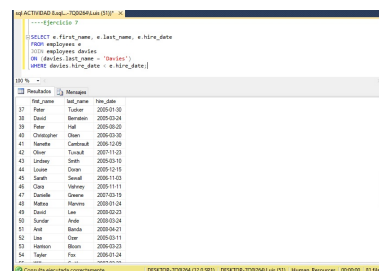
Row	ID_Empleado	Empleado	ID_Manager	Manager
1	100	De Haan	100	King
2	101	De Haan	100	King
3	102	De Haan	100	King
4	103	De Haan	100	King
5	104	De Haan	100	King
6	105	De Haan	100	King
7	106	De Haan	100	King
8	107	De Haan	100	King
9	108	De Haan	100	King
10	109	De Haan	100	King
11	110	De Haan	100	King
12	111	De Haan	100	King
13	112	De Haan	100	King
14	113	De Haan	100	King
15	114	De Haan	100	King
16	115	De Haan	100	King
17	116	De Haan	100	King
18	117	De Haan	100	King
19	118	De Haan	100	King
20	119	De Haan	100	King

6. Crear un reporte que muestre los No de Departamento y Apellidos de todos los empleados, asimismo adicionar una columna con los Apellidos de todos empleados que trabajan en el mismo

```
select e.department_id 'DEPARTAMENTO', e.last_name 'EMPLEADO',
d.last_name 'COLEGA'
from employees e
join employees d
on (e.department_id=d.department_id) and e.last_name!=d.last_name;
```



- ```
SELECT e.first_name, e.last_name, e.hire_date
FROM employees e
JOIN employees davies
ON (davies.last_name = 'Davies')
WHERE davies.hire_date < e.hire_date;
```



- ```
select e.last_name 'EMPLEADO', e.hire_date 'FECHA_CONTRATACION' , j.last_name 'ADMINISTRADOR',
j.hire_date 'FECHA_CONTRATACION_ADMINISTRADOR'
from employees e
join employees j
on e.manager_id=j.employee_id
and e.hire_date < j.hire_date
order by e.hire_date;
```

26

9. Actividad No 09 – SubConsultas

1. El departamento de Recursos Humanos requiere una consulta que pregunte al usuario por el Apellido del empleado, Luego la consulta deberá mostrar los Apellidos y Fecha de Contratación de todos los empleados del mismo departamento excluyendo o con excepción del empleado el cual ha sido proporcionado su apellido reporte que muestre las direcciones de todos los departamentos.

– leyendo id de empleado

```
SET @empid 110
```

– obteniendo id de departamento de empleado

```
SET @depid (SELECT emp.department_id
```

```
FROM employees as emp
```

```
WHERE emp.employee_id=@empid);
```

– todos los empleados del mismo departamento excluyendo al empleado ingresado anteriormente

```
SELECT emp.employee_id,
```

```
emp.last_name,
```

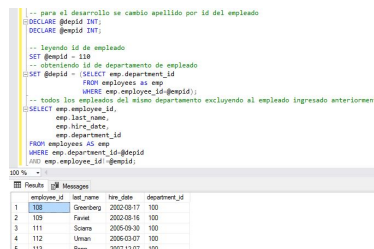
```
emp.hire_date,
```

```
emp.department_id
```

```
FROM employees AS emp
```

```
WHERE emp.department_id=@depid
```

```
AND emp.employee_id!@empid;
```



```
-- para el desarrollo se cambio apellido por id del empleado
--DECLARE @depid INT;
--DECLARE @empid INT;

-- leyendo id de empleado
SET @empid = 110
-- obteniendo id de departamento de empleado
SET @depid = (SELECT emp.department_id
FROM employees as emp
WHERE emp.employee_id=@empid);
-- todos los empleados del mismo departamento excluyendo al empleado ingresado anteriormente
SELECT emp.employee_id,
emp.last_name,
emp.hire_date,
emp.department_id
FROM employees AS emp
WHERE emp.department_id=@depid
AND emp.employee_id!@empid;
```

employee_id	last_name	hire_date	department_id
108	Chen	2002-08-17	100
109	Favre	2002-08-16	100
111	Schaer	2005-09-30	100
112	Ullrich	2006-03-07	100
113	Popp	2007-12-07	100

2. Crear un reporte que muestre el No del Empleado, Apellidos y Salarios de todos los empleados que tienen un salario superior al promedio de salarios de todos los empleados. Ordenar los resultados por el Salario de forma ascendente.

```
SELECT emp.employee_id,
```

```
emp.last_name,
```

```
emp.salary
```

```
FROM employees AS emp
```

```
WHERE emp.salary > @prom;
```

```

-- Se considera "Nº de empleado" como "id de empleado"
-- Obteniendo promedio de salario
DECLARE @prom DECIMAL(8,2); --Variable promedio
SET @prom = (SELECT AVG(salary) FROM employees);

-- Todos los empleados con sueldo superior al promedio
SELECT emp.employee_id,
       emp.last_name,
       emp.salary
FROM employees AS emp
WHERE emp.salary > @prom;

```

employee_id	last_name	salary
100	King	24000.00
101	Kochhar	17000.00
102	De Haan	17000.00
103	Hunold	9000.00
108	Greenberg	12008.00
109	Faviet	9000.00
110	Chen	8200.00
111	Sciarra	7700.00
112	Uman	7800.00
113	Popp	6900.00

3. Realizar un reporte que muestre el No de Empleado y Apellidos de todos los empleados quienes trabajan en el departamento de cualquier empleado que su apellido contenga la letra 'u'.

```

SELECT emp.employee_id,
       emp.last_name,
       emp.department_id
FROM employees AS emp
JOIN (SELECT DISTINCT department_id
      FROM employees
      WHERE last_name LIKE '%u%') AS depid
ON emp.department_id = depid.department_id;

```

```

-- Se considera "Nº de empleado" como "id de empleado"
-- Obtener los id de departamentos de los empleados que contengan "u" en su apellido
SELECT DISTINCT department_id
FROM employees
WHERE last_name LIKE '%u%';

-- Obtener todos los empleados que laboren en alguno de los departamentos hallados anteriormente
SELECT emp.employee_id,
       emp.last_name,
       emp.department_id
FROM employees AS emp
JOIN (SELECT DISTINCT department_id
      FROM employees
      WHERE last_name LIKE '%u%') AS depid
ON emp.department_id = depid.department_id;

```

employee_id	last_name	department_id
114	Patney	30
115	Abene	30
116	Bede	30
117	Toben	30
118	Hunum	30
119	Colmanese	30
120	Tajer	30
121	Reaur	30
122	Sullivan	30
123	Geert	30

4. El departamento de Recursos Humanos requiere un reporte que muestre los Apellidos, No de Departamento y Puestos de los empleados cuya locación de departamento es 1700.

```

SELECT emp.last_name,
       emp.department_id,
       dep.location_id
FROM employees as emp
JOIN departments as dep
ON emp.department_id = dep.department_id
WHERE dep.location_id = 1700;

```

```
-- 4
SELECT emp.last_name,
       emp.department_id,
       dep.location_id
FROM employees as emp
JOIN departments as dep
ON emp.department_id=dep.department_id
WHERE dep.location_id=1700;
```

	last_name	department_id	location_id
1	King	90	1700
2	Kochhar	90	1700
3	De Haan	90	1700
4	Greenberg	100	1700
5	Faviet	100	1700
6	Chen	100	1700
7	Sciarra	100	1700
8	Uman	100	1700
9	Popp	100	1700
10	Raphaely	30	1700

5. Modificar la consulta anterior de forma que el usuario pueda introducir el No de locación.

```
DECLARE @locid INT;
SET @locid = 1700;
SELECT emp.last_name,
       emp.department_id,
       dep.location_id
FROM employees as emp
JOIN departments as dep
ON emp.department_id=dep.department_id
WHERE dep.location_id=@locid;
```

```
-- 5
DECLARE @locid INT;
SET @locid = 1700;
SELECT emp.last_name,
       emp.department_id,
       dep.location_id
FROM employees as emp
JOIN departments as dep
ON emp.department_id=dep.department_id
WHERE dep.location_id=@locid;
```

	last_name	department_id	location_id
1	King	90	1700
2	Kochhar	90	1700
3	De Haan	90	1700
4	Greenberg	100	1700
5	Faviet	100	1700
6	Chen	100	1700
7	Sciarra	100	1700
8	Uman	100	1700
9	Popp	100	1700
10	Raphaely	30	1700

6. Crear un reporte para el departamento de Recursos Humanos que muestre los Apellidos y Salarios de todos los empleados cuyo Administrador apellido 'King'.

```
SELECT emp.last_name,
       emp.salary
FROM employees AS emp
JOIN (SELECT dep.department_id
FROM departments AS dep
JOIN (SELECT employee_id,
```

```

last_name
FROM employees
WHERE last_name='KING') AS manking
ON dep.manager_idmanking.employee_id) AS depking
ON emp.department_iddepking.department_id;

```

```

-- 6
-- conseguir id de empleado que lleven como apellido KING
SELECT employee_id,
       last_name
FROM employees
WHERE last_name='KING';
-- conseguir id de departamentos que coincidan en manager_id con employee_id
SELECT dep.department_id
FROM departments AS dep
JOIN (SELECT employee_id,
       last_name
      FROM employees
      WHERE last_name='KING') AS manking
ON dep.manager_id=manking.employee_id
-- FINALMENTE, apellidos y salarios de empleados que tengan como id de
-- departamento el/los id de departamentos hallados anteriormente
SELECT emp.last_name,
       emp.salary
FROM employees AS emp
JOIN (SELECT dep.department_id
      FROM departments AS dep
      JOIN (SELECT employee_id,
                  last_name
              FROM employees
              WHERE last_name='KING') AS manking
      ON dep.manager_id=manking.employee_id) AS depking
ON emp.department_id=depking.department_id;

```

last_name	salary
King	24000.00
Kochhar	17000.00
De Haan	17000.00

7. Crear un reporte para el departamento de Recursos Humanos que muestre el No de Departamento, Apellidos, Puestos de todos los empleados en el departamento 'Executive'.

```

SELECT empnomjob.department_id,
empnomjob.last_name,
empnomjob.job_title
FROM departments
JOIN (SELECT emp.department_id,
emp.last_name,
jobs.job_title
FROM employees AS emp
JOIN jobs
ON emp.job_id=jobs.job_id) AS empnomjob
ON empnomjob.department_id=departments.department_id
WHERE department_name='executive'

```

```

-- 7
select * from employees where department_id=90;
select * from jobs;
select * from departments where department_name='executive';
-- consiguiendo empleados con nombre de puesto
SELECT emp.department_id,
       emp.last_name,
       jobs.job_title
FROM employees AS emp
JOIN jobs
ON emp.job_id=jobs.job_id;
-- FINALMENTE, empleados con departamento Executive
SELECT empnomjob.department_id,
       empnomjob.last_name,
       empnomjob.job_title
FROM departments
JOIN (SELECT emp.department_id,
       emp.last_name,
       jobs.job_title
      FROM employees AS emp
      JOIN jobs
      ON emp.job_id=jobs.job_id) AS empnomjob
ON empnomjob.department_id=departments.department_id
WHERE department_name='executive';

```

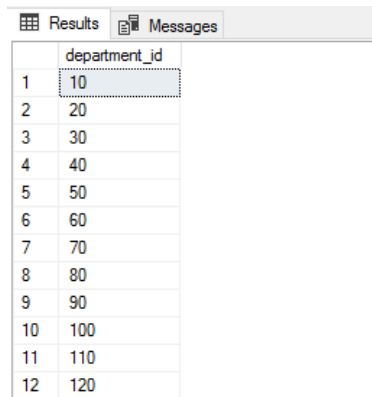
department_id	last_name	job_title
90	King	President
90	Kochhar	Administration Vice President
90	De Haan	Administration Vice President

8. Modificar la consulta del ítem 4.3 para que adicionalmente se muestro solo a los empleados que tengan un salario mayor al promedio de todos los salarios de los empleados.

10. Actividad No 10 – Conjuntos

1. El departamento de Recursos Humanos requiere un reporte de todos los departamentos que no contengan un empleado con el puesto 'ST_CLERK'. Utilizar el operador MINUS o EXCEPT para esta solicitud.

```
select department_id from employees
where job_id ='ST_CLERK';
```

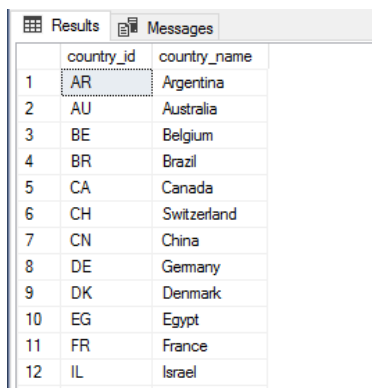


The screenshot shows a database query result with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'department_id' and an index from 1 to 12. The values in the 'department_id' column are 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, and 120. The first row (index 1) is highlighted with a dashed border.

	department_id
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100
11	110
12	120

2. El departamento de Recursos Humanos requiere adicionalmente una lista de todos los países que no tengan un departamento de la empresa localizado en ellos, mostrar el código del país y el nombre. Utilizar el operador MINUS o EXCEPT para realizar esta operación.

```
select country_id, country_name from countries
minus
```



The screenshot shows a database query result with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'country_id' and 'country_name'. The values in the 'country_id' column are AR, AU, BE, BR, CA, CH, CN, DE, DK, EG, FR, and IL. The corresponding country names are Argentina, Australia, Belgium, Brazil, Canada, Switzerland, China, Germany, Denmark, Egypt, France, and Israel. The first row (index 1) is highlighted with a dashed border.

	country_id	country_name
1	AR	Argentina
2	AU	Australia
3	BE	Belgium
4	BR	Brazil
5	CA	Canada
6	CH	Switzerland
7	CN	China
8	DE	Germany
9	DK	Denmark
10	EG	Egypt
11	FR	France
12	IL	Israel

3. Se necesita una lista de puestos de los departamentos 10, 50 y 20, en ese orden, mostrar el código del puesto y código del departamento. Utilizar el operador UNION ALL.

```
select distinct job_id, department_id from employees
where (department_id=10)
union
select distinct job_id, department_id from employees
where (department_id=50)
```

```
union
select distinct job_id, department_id from employees
where (department_id=20);
```

	job_id	department_id
1	AD_ASST	10
2	MK_MAN	20
3	MK_REP	20

4. Crear un reporte que muestre que liste los códigos de los empleados y los puestos de todos aquellos empleados que tienen el mismo puesto que en el momento en el que fueron contratados por la empresa, cambiaron de puestos y luego volvieron al puesto anterior. Utilizar el operador INTERSECT.

```
select employee_id, job_id from employees
intersect
select distinct employee_id, job_id from job_history;
```

	employee_id	job_id
1	176	SA_REP
2	200	AD_ASST

5. El departamento de Recursos Humanos requiere un reporte que muestre lo siguiente:
 - Apellidos y códigos de departamentos de todos los registros de la tabla empleados sin importar si pertenecen a uno o ningún departamento.
 - Código de departamentos y nombres de departamentos de la tabla DEPARTAMENTOS inclusive si no existiese ningún empleado en ese departamento

Ambos requerimientos se deben mostrar en un mismo resultado. Utilizar el operador UNION ALL.

```
select last_name, department_id, null from employees union select null, department_id, department_name from departments;
```

	last_name	department_id	(No column name)
1	NULL	10	Administration
2	NULL	20	Marketing
3	NULL	30	Purchasing
4	NULL	40	Human Resources
5	NULL	50	Shipping
6	NULL	60	IT
7	NULL	70	Public Relations
8	NULL	80	Sales
9	NULL	90	Executive
10	NULL	100	Finance
11	NULL	110	Accounting
12	NULL	120	Treasury