

UNERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**INFORME DE LABORATORIO N 01**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Cuadros Quiroga

Integrantes:

Layme Valeriano Diego Rolando	(2017057865)
Yonathan Mamani Calisaya	(2017057863)
Moreno Mulluni Luis Angel	(2017057864)
Rodriguez Mamani Juanl	(2017057862)
Taquila Carazas Percy Huber	(2018061088)
Apaza Mamani Edward	(2018060915)

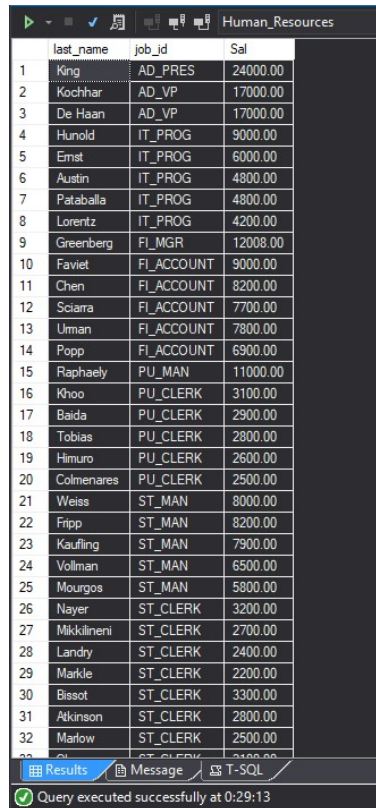
# Índice

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	7
5. Actividad No 05 – Funciones	12
6. Actividad No 06 – Funciones de Conversión	15
7. Actividad No 07 – Funciones de Agrupación	18
8. Actividad No 08 – Enlaces	21
9. Actividad No 09 – SubConsultas	24
10. Actividad No 10 – Conjuntos	28

# 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?

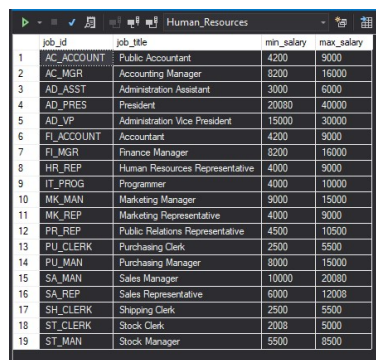
1. SELECT last\_name, job\_id, salary AS Sal FROM employees;  
Es correcta



The screenshot shows a SQL query window titled 'Human\_Resources' with the following query: `SELECT last_name, job_id, salary AS Sal FROM employees;` The results are displayed in a table with 32 rows. The status bar at the bottom indicates 'Query executed successfully at 0:29:13'.

	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_ACCOUNT	9000.00
11	Chen	FI_ACCOUNT	8200.00
12	Sciarra	FI_ACCOUNT	7700.00
13	Urman	FI_ACCOUNT	7800.00
14	Popp	FI_ACCOUNT	6900.00
15	Raphaely	PU_MAN	11000.00
16	Khoo	PU_CLERK	3100.00
17	Baida	PU_CLERK	2900.00
18	Tobias	PU_CLERK	2800.00
19	Himuro	PU_CLERK	2600.00
20	Colmenares	PU_CLERK	2500.00
21	Weiss	ST_MAN	8000.00
22	Frippe	ST_MAN	8200.00
23	Kaufling	ST_MAN	7900.00
24	Vollman	ST_MAN	6500.00
25	Mourgos	ST_MAN	5900.00
26	Nayer	ST_CLERK	3200.00
27	Mikkilineni	ST_CLERK	2700.00
28	Landry	ST_CLERK	2400.00
29	Markle	ST_CLERK	2200.00
30	Bissot	ST_CLERK	3300.00
31	Atkinson	ST_CLERK	2800.00
32	Marlow	ST_CLERK	2500.00

2. SELECT \* FROM job\_grades;  
Es incorrecta, la sentencia correcta sería:  
SELECT \* FROM jobs;



The screenshot shows a SQL query window titled 'Human\_Resources' with the following query: `SELECT * FROM jobs;` The results are displayed in a table with 19 rows. The status bar at the bottom indicates 'Query executed successfully at 0:29:13'.

	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 President	15000	30000
6	FI_ACCOUNT	Accountant	4200	9000
7	FI_MGR	Finance Manager	8200	16000
8	HR_REP	Human Resources Representative	4000	9000
9	IT_PROG	Programmer	4000	10000
10	MK_MAN	Marketing Manager	9000	15000
11	MK_REP	Marketing Representative	4000	9000
12	PR_REP	Public Relations Representative	4500	10500
13	PU_CLERK	Purchasing Clerk	2500	5500
14	PU_MAN	Purchasing Manager	8000	15000
15	SA_MAN	Sales Manager	10000	20080
16	SA_REP	Sales Representative	6000	12008
17	SH_CLERK	Shipping Clerk	2500	5500
18	ST_CLERK	Stock Clerk	2008	5000
19	ST_MAN	Stock Manager	5500	8500

3. 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;

Human_Resources			
	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	Greenberg	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
16	115	Khoo	37200.00
17	116	Baida	34800.00
18	117	Tobias	33600.00
19	118	Himuro	31200.00
20	119	Colmenares	30000.00
21	120	Weiss	96000.00
22	121	Fripp	98400.00
23	122	Kaufling	94800.00
24	123	Vollman	78000.00
25	124	Mourgos	69600.00
26	125	Nayer	38400.00
27	126	Mikkilineni	32400.00
28	127	Landry	28800.00
29	128	Markle	26400.00
30	129	Bissot	39600.00
31	130	Atkinson	33600.00
32	131	Marlow	30000.00
33	132	Chen	265600.00

Results Message T-SQL  
 Query executed successfully at 0:36:54

## 2. Actividad No 02 – Reconociendo la estructura

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

SP\_HELP 'DEPARTMENTS'

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
100	King	AD_PRES	2003-06-17
101	Kochhar	AD_VP	2005-09-21
102	De Haan	AD_VP	2001-01-13
103	Hunold	IT_PROG	2006-01-03
104	Ernst	IT_PROG	2007-05-21
105	Austin	IT_PROG	2005-06-25
106	Pataballa	IT_PROG	2006-02-05
107	Lorentz	IT_PROG	2007-02-07
108	Greenberg	FI_MGR	2002-08-17
109	Faviet	FI_ACCOUNT	2002-08-16
110	Chen	FI_ACCOUNT	2005-09-28
111	Sciarra	FI_ACCOUNT	2005-09-30
112	Uman	FI_ACCOUNT	2006-03-07
113	Popp	FI_ACCOUNT	2007-12-07
114	Raphaely	PU_MAN	2002-12-07
115	Khoo	PU_CLERK	2003-05-18
116	Baida	PU_CLERK	2005-12-24
117	Tobias	PU_CLERK	2005-07-24
118	Himuro	PU_CLERK	2006-11-15
119	Colmenares	PU_CLERK	2007-08-10
120	Weiss	ST_MAN	2004-07-18
121	Fripp	ST_MAN	2005-04-10
122	Kaufling	ST_MAN	2003-05-01
123	Vollman	ST_MAN	2005-10-10
124	Mourgos	ST_MAN	2007-11-16
125	Nayer	ST_CLERK	2005-07-16
126	Mikkilineni	ST_CLERK	2006-09-28
127	Landry	ST_CLERK	2007-01-14
128	Markle	ST_CLERK	2008-03-08
129	Bissot	ST_CLERK	2005-08-20
130	Atkinson	ST_CLERK	2005-10-30
131	Marlow	ST_CLERK	2005-02-16

Query executed successfully at 0:31:05

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.

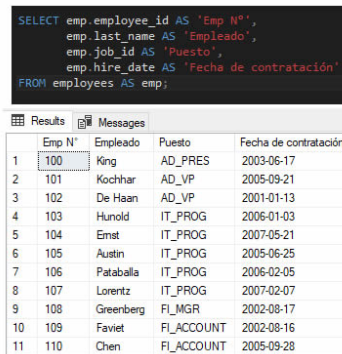
```
SELECT DISTINCT job_id FROM employees;
```

job_id
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN
MK_REP
PR_REP
PU_CLERK
PU_MAN
SA_MAN
SA_REP
SH_CLERK
ST_CLERK
ST_MAN

### 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;
```



The screenshot shows a SQL query window with the following query:

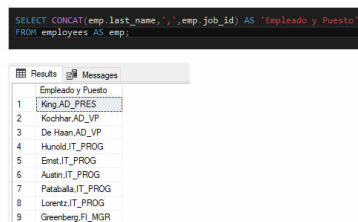
```
SELECT emp.employee_id AS 'Emp N',  
emp.last_name AS 'Empleado',  
emp.job_id AS 'Puesto',  
emp.hire_date AS 'Fecha de contratación'  
FROM employees AS emp;
```

Below the query window, the 'Results' tab is active, displaying a table with 4 columns: 'Emp N', 'Empleado', 'Puesto', and 'Fecha de contratación'. The table contains 11 rows of data.

	Emp N	Empleado	Puesto	Fecha de contratación
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

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;
```



The screenshot shows a SQL query window with the following query:

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

Below the query window, the 'Results' tab is active, displaying a table with 1 column: 'Empleado y Puesto'. The table contains 9 rows of data.

Empleado y Puesto
King,AD_PRES
Kochhar,AD_VP
De Haan,AD_VP
Hunold,IT_PROG
Ernst,IT_PROG
Austin,IT_PROG
Pataballa,IT_PROG
Lorentz,IT_PROG
Greenberg,FI_MGR

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;

```

```

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;

```

Results Messages

Los 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,Faviet,DFAVIET,515.124.4169,2002-08-16,FI...
11	110,John,Chen,JCHEN,515.124.4269,2005-09-28,FI_ACC...
12	111,Jamuel,Sciarra,JSICIARRA,515.124.4369,2005-09-30,FI...



## 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;`

```
SELECT last_name,salary FROM employees WHERE salary > 12000;  
GO
```

	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;`

```
SELECT last_name,department_id FROM employees WHERE employee_id < 176;  
GO
```

	last_name	department_id
1	Livingston	80
2	Grant	NULL
3	Johnson	80
4	Taylor	50
5	Plesur	50
6	Sullivan	50
7	Geoni	50
8	Sandhand	50
9	Bull	50
10	Dellinger	50
11	Cabrio	50
12	Chung	50

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;`

```
SELECT last_name,job_id,salary as Sal FROM employees WHERE salary < 5000 AND salary > 12000;  
GO
```

	last_name	job_id	Sal
1	Hunold	IT_PROG	9000.00
2	Ernst	IT_PROG	6000.00
3	Faviet	PL_ACCOUNT	9000.00
4	Chen	PL_ACCOUNT	8200.00
5	Scarra	PL_ACCOUNT	7700.00
6	Ullman	PL_ACCOUNT	7800.00
7	Poo	PL_ACCOUNT	6900.00
8	Raphaely	PL_MGR	11000.00
9	Whens	ST_MGR	8000.00
10	Poo	ST_MGR	8200.00
11	Kauffman	ST_MGR	7900.00
12	Vollman	ST_MGR	6900.00

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;`

```
SELECT last_name,job_id,hire_date FROM employees
WHERE last_name = 'Matos' OR last_name = 'Taylor' ORDER BY hire_date ASC;
GO
```

	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,department_id FROM employees
WHERE department_id = 20 OR department_id = 50 ORDER BY last_name ASC;
GO
```

	last_name	department_id
1	Atkinson	50
2	Bell	50
3	Blossot	50
4	Bull	50
5	Cabrio	50
6	Chung	50
7	Davies	50
8	Dellinger	50
9	Dilly	50
10	Everett	50
11	Fay	20
12	Feeney	50

6. Modificar el reporte del ítem 4.1. para mostrar los apellidos y salarios de los empleados que tengan un salario entre los \$ 5,000 a \$ 12,000 y pertenezcan a los números de departamento 20 o 50. Asimismo etiquetar las cabeceras de los resultados con los alias Empleado y Salario Mensual respectivamente.

select last\_name 'Empleado',salary 'Salario Mensual' from employees where salary >5000 and salary <12000 and (department\_id = 20 or department\_id = 50);

```
SELECT last_name 'Empleado',salary 'Salario Mensual'
FROM employees WHERE salary > 5000 AND salary < 12000
AND (department_id = 20 OR department_id = 50);
GO
```

	last_name	department_id
1	Atkinson	50
2	Bell	50
3	Blossot	50
4	Bull	50
5	Cabrio	50
6	Chung	50
7	Davies	50
8	Dellinger	50
9	Dilly	50
10	Everett	50
11	Fay	20
12	Feeney	50

7. El departamento de Recursos Humanos necesita un listado de apellidos (last\_name) y fecha de contratación (hire\_date) de todos los empleados que fueron contratados el año 1994.

select last\_name,hire\_date from employees where hire\_date between '19940101' and '19941231';

```
SELECT last_name,hire_date FROM employees WHERE hire_date = '1994'
GO
```

	last_name	hire_date
--	-----------	-----------

8. Crear un reporte que muestre los apellidos (last\_name) y puesto (job\_id) de todos los empleados que no tengan un administrador (manager).

```
select last_name,job_id from employees where manager_id is null;
```

```
SELECT last_name,job_id FROM employees WHERE manager_id IS NULL;
GO
```

	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;
```

```
SELECT last_name,salary,commission_pct FROM employees
ORDER BY salary DESC,commission_pct DESC;
GO
```

	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	Ernazuiz	12000.00	0.30
10	Ozer	11500.00	0.25
11	Cambraut	11000.00	0.30
12	Abel	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;
```

```
DECLARE @salario AS DECIMAL(9,2);
SET @salario = 12000;
SELECT last_name,salary FROM employees WHERE salary > @salario;
GO
```

	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	Ernazuiz	12000.00	0.30
10	Ozer	11500.00	0.25
11	Cambraut	11000.00	0.30
12	Abel	11000.00	0.30

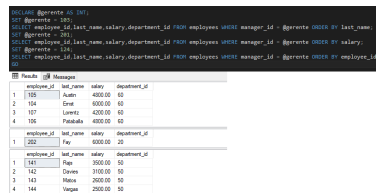
11. El departamento de Recursos Humanos requiere extraer reporte basados en el Administrador (manager\_id). Se requiere crear una consulta que pregunte al usuario por el Administrador (manager\_id) y genere un reporte con los números de empleado (employee\_id), apellidos (last\_name), salarios (salary) 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
```



The screenshot shows three separate query results for different manager IDs. Each result is a table with columns: employee\_id, last\_name, salary, and department\_id.

employee_id	last_name	salary	department_id
1	100	4000.00	60
2	101	3000.00	60
3	102	2000.00	60
4	103	1000.00	60

employee_id	last_name	salary	department_id
1	200	4000.00	20

employee_id	last_name	salary	department_id
1	124	2000.00	50
2	125	3000.00	50
3	126	4000.00	50
4	127	5000.00	50

12. 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
```

```
SELECT last_name FROM employees
WHERE SUBSTRING(last_name,3,1) = 'a'
GO
```

Results Messages

	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
```

```
SELECT last_name FROM employees
WHERE SUBSTRING(last_name,3,1) = 'e' OR SUBSTRING(last_name,3,1) = 'E';
```

	last_name
1	Greenberg
2	Chen
3	Gee
4	McEwen
5	Greene
6	Lee
7	Ozer
8	Abel
9	Grant
10	Plesner
11	Everett
12	Feesey

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
```

```
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 sa-
lary >5000 and salary <12000 and (department_id = 20 or department_id = 50) and commis-
sion_pct = 0.20;
go
```

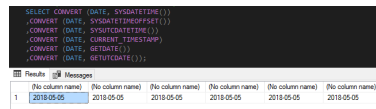
```
SELECT last_name 'Empleado',salary 'Salario Mensual',commission_pct
FROM employees WHERE salary > 5000 AND salary < 12000
AND (department_id = 20 OR department_id = 50) AND commission_pct = 0.20;
GO
```

	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());
```



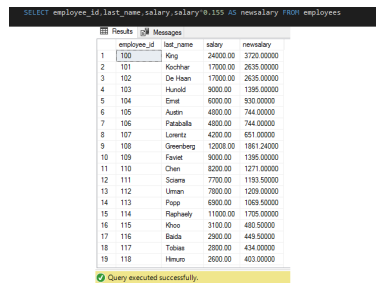
The screenshot shows a SQL Server query window with the following SQL code:

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

The results pane shows a single row of data with six columns, each containing a date and time value converted from the current system time.

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
```



The screenshot shows a SQL Server query window with the following SQL code:

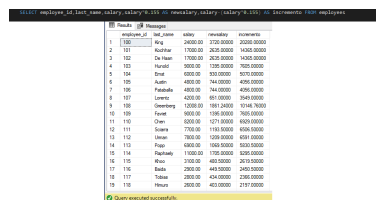
```
SELECT employee_id,last_name,salary,salary*0.155 as newsalary FROM employees
```

The results pane shows a table with four columns: employee\_id, last\_name, salary, and newsalary. The data is as follows:

employee_id	last_name	salary	newsalary
100	King	24000.00	3720.00000
101	Neer	17000.00	2635.00000
102	De Haan	17000.00	2635.00000
103	Hunold	9000.00	1395.00000
104	Ernst	6000.00	930.00000
105	Austin	4800.00	744.00000
106	Pataballa	4800.00	744.00000
107	Lorentz	4200.00	651.00000
108	Gawberg	12000.00	1860.00000
109	Faviet	9000.00	1395.00000
110	Chen	8200.00	1271.00000
111	Scott	7700.00	1193.00000
112	Uman	7800.00	1209.00000
113	Popp	6900.00	1069.50000
114	Pyburn	11000.00	1705.00000
115	Rhys	3100.00	480.50000
116	Baida	2900.00	449.50000
117	Tobias	2800.00	434.00000
118	Hem	2600.00	401.00000

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
```



The screenshot shows a SQL Server query window with the following SQL code:

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

The results pane shows a table with five columns: employee\_id, last\_name, salary, newsalary, and incremento. The data is as follows:

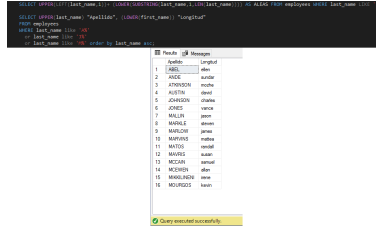
employee_id	last_name	salary	newsalary	incremento
100	King	24000.00	3720.00000	20280.00000
101	Neer	17000.00	2635.00000	14365.00000
102	De Haan	17000.00	2635.00000	14365.00000
103	Hunold	9000.00	1395.00000	7605.00000
104	Ernst	6000.00	930.00000	5070.00000
105	Austin	4800.00	744.00000	4056.00000
106	Pataballa	4800.00	744.00000	4056.00000
107	Lorentz	4200.00	651.00000	3549.00000
108	Gawberg	12000.00	1860.00000	10140.00000
109	Faviet	9000.00	1395.00000	7605.00000
110	Chen	8200.00	1271.00000	6929.00000
111	Scott	7700.00	1193.00000	6507.00000
112	Uman	7800.00	1209.00000	6591.00000
113	Popp	6900.00	1069.50000	5830.50000
114	Pyburn	11000.00	1705.00000	9295.00000
115	Rhys	3100.00	480.50000	2619.50000
116	Baida	2900.00	449.50000	2450.50000
117	Tobias	2800.00	434.00000	2366.00000
118	Hem	2600.00	401.00000	2199.00000

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;

```



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;

```

```
SELECT last_name, lpad(salary,15,'$') as salary from employees;
```

	VALUE
1	\$\$\$\$\$\$24000.00
2	\$\$\$\$\$\$17000.00
3	\$\$\$\$\$\$17000.00
4	\$\$\$\$\$\$99000.00
5	\$\$\$\$\$\$66000.00
6	\$\$\$\$\$\$48000.00
7	\$\$\$\$\$\$48000.00
8	\$\$\$\$\$\$42000.00
9	\$\$\$\$\$\$12000.00
10	\$\$\$\$\$\$99000.00
11	\$\$\$\$\$\$82000.00
12	\$\$\$\$\$\$77000.00
13	\$\$\$\$\$\$78000.00
14	\$\$\$\$\$\$69000.00
15	\$\$\$\$\$\$11000.00
16	\$\$\$\$\$\$31000.00
17	\$\$\$\$\$\$29000.00
18	\$\$\$\$\$\$28000.00
19	\$\$\$\$\$\$26000.00

Query executed successfully.

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.

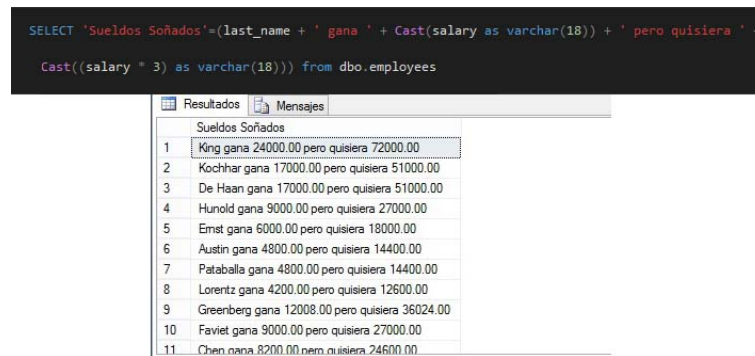
```
select last_name, TRUNC(((SYSDATE-hire_date)/7),0) as TENURE from employees where department_id=90 ORDER BY hire_date DESC;
```



## 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
```



The screenshot shows a SQL Server query window with the following query:

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

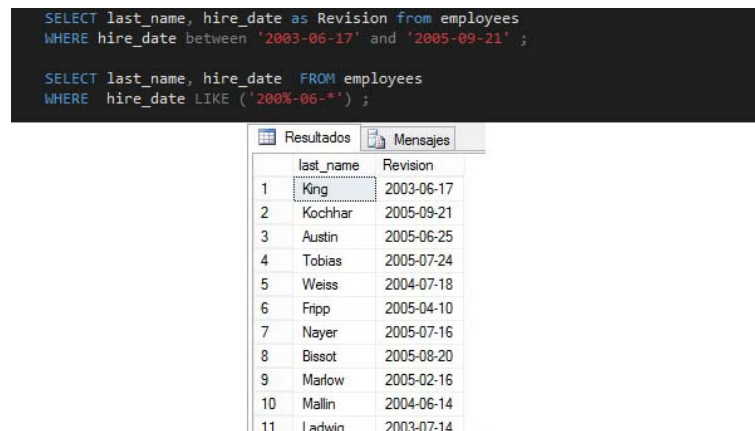
The results pane shows a table with 11 rows. The first row is highlighted:

	Sueldos Soñados
1	King gana 24000.00 pero quisiera 72000.00
2	Kochhar gana 17000.00 pero quisiera 51000.00
3	De Haan gana 17000.00 pero quisiera 51000.00
4	Hunold gana 9000.00 pero quisiera 27000.00
5	Ernst gana 6000.00 pero quisiera 18000.00
6	Austin gana 4800.00 pero quisiera 14400.00
7	Pataballa gana 4800.00 pero quisiera 14400.00
8	Lorentz gana 4200.00 pero quisiera 12600.00
9	Greenberg gana 12008.00 pero quisiera 36024.00
10	Faviet gana 9000.00 pero quisiera 27000.00
11	Chen gana 8200.00 pero quisiera 24600.00

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
```



The screenshot shows a SQL Server query window with the following query:

```
SELECT last_name, hire_date as Revision from employees  
WHERE hire_date between '2003-06-17' and '2005-09-21';  
  
SELECT last_name, hire_date FROM employees  
WHERE hire_date LIKE ('200%-06-*');
```

The results pane shows a table with 11 rows. The first row is highlighted:

	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
11	Ladwin	2003-07-14

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

```

```

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

```

	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	Kauffman	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

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

```

```

--muestras texto comision, si no tiene monto
SELECT last_name AS 'Apellidos', 'Comision'='Sin Comision' FROM dbo.employees WHERE commission_pct <= 0
UNION
--muestra comision y el calculo de dicha
SELECT last_name as 'Apellidos', 'Comision'= Cast((salary * commission_pct) as varchar(20)) from dbo.employees

```

	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	Cambrault	1500.0000
8	Cambrault	3300.0000
9	Doran	2250.0000
10	Errazuriz	3600.0000
11	Fox	1920.0000

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
```

```
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'
WHEN j.job_id = 'ST_CLERK' THEN 'E'
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	Emst	Programmer	C
6	Austin	Programmer	C
7	Pataballa	Programmer	C
8	Lorentz	Programmer	C
9	Greenberg	Finance Manager	0
10	Faviet	Accountant	0
11	Chen	Accountant	0

## 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;
```

Results		Messages		
	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.

```
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 GROUP BY employees.job_id;
```

```
--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 GROUP BY employees.job_id;
```

	Maximo	Minimo	Sumatoria	Promedio
1	8500.00	1300.00	1300.00	8500.000000
2	12000.00	12000.00	12000.00	12000.000000
3	4400.00	4400.00	4400.00	4400.000000
4	24000.00	24000.00	24000.00	24000.000000
5	17000.00	17000.00	17000.00	17000.000000
6	9000.00	6500.00	25600.00	7600.000000
7	12000.00	12000.00	12000.00	12000.000000
8	6500.00	6500.00	6500.00	6500.000000
9	6500.00	4200.00	20500.00	5760.000000
10	13000.00	13000.00	13000.00	13000.000000
11	6000.00	6000.00	6000.00	6000.000000
12	10000.00	10000.00	10000.00	10000.000000
13	3100.00	2000.00	13000.00	2750.000000
14	11000.00	11000.00	11000.00	11000.000000
15	4000.00	10000.00	41000.00	12300.000000
16	11000.00	6100.00	25000.00	8300.000000
17	4200.00	2500.00	44000.00	3215.000000
18	3600.00	2100.00	55700.00	2755.000000
19	6200.00	5800.00	36400.00	7280.000000

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;
```

Results Messages	
(No column name)	
1	1
2	1
3	1
4	1
5	2
6	5
7	1
8	1
9	5
10	1
11	1
12	1
13	5
14	1
15	5
16	30

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;
```

Results Messages	
Número de Administradores	
1	18

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;
```

Results Messages	
diferencia	
1	21900.00

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	6900.00	108
7	7000.00	145
8	7000.00	146
9	8300.00	205
10	9000.00	102
11	24000.00	NULL

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.

SELECT count(\*) as totalempleados , (SELECT count(\*)  
 from employees  
 where hire\_date between '1995-01-01' and '1999-01-01') AS PORAÑO  
 from employees

<pre> SELECT count(*) as totalempleados , (SELECT count(*) from employees where hire_date between '1995-01-01' and '1999-01-01') AS PORAÑO from employees </pre>	
Results	Messages
1	107 0

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.

SELECT  
 e.employee\_id 'ID Empleado', j.job\_title 'Nombre trabajo', e.salary 'Salario', e.department\_id 'Nro Departamento'  
 from jobs as j  
 join employees as e  
 on j.job\_id = e.job\_id

<pre> SELECT e.employee_id 'ID Empleado', j.job_title 'Nombre trabajo', e.salary 'Salario', e.department_id 'Nro Departamento' from jobs as j join employees as e on j.job_id = e.job_id </pre>			
Results			
1	102	102	102
2	102	102	102
3	102	102	102
4	102	102	102
5	102	102	102
6	102	102	102
7	102	102	102
8	102	102	102
9	102	102	102
10	102	102	102
11	102	102	102
12	102	102	102
13	102	102	102
14	102	102	102
15	102	102	102
16	102	102	102
17	102	102	102
18	102	102	102
19	102	102	102
20	102	102	102
21	102	102	102
22	102	102	102
23	102	102	102
24	102	102	102
25	102	102	102
26	102	102	102
27	102	102	102
28	102	102	102
29	102	102	102
30	102	102	102
31	102	102	102
32	102	102	102
33	102	102	102
34	102	102	102
35	102	102	102
36	102	102	102
37	102	102	102
38	102	102	102
39	102	102	102
40	102	102	102
41	102	102	102
42	102	102	102
43	102	102	102
44	102	102	102
45	102	102	102
46	102	102	102
47	102	102	102
48	102	102	102
49	102	102	102
50	102	102	102
51	102	102	102
52	102	102	102
53	102	102	102
54	102	102	102
55	102	102	102
56	102	102	102
57	102	102	102
58	102	102	102
59	102	102	102
60	102	102	102
61	102	102	102
62	102	102	102
63	102	102	102
64	102	102	102
65	102	102	102
66	102	102	102
67	102	102	102
68	102	102	102
69	102	102	102
70	102	102	102
71	102	102	102
72	102	102	102
73	102	102	102
74	102	102	102
75	102	102	102
76	102	102	102
77	102	102	102
78	102	102	102
79	102	102	102
80	102	102	102
81	102	102	102
82	102	102	102
83	102	102	102
84	102	102	102
85	102	102	102
86	102	102	102
87	102	102	102
88	102	102	102
89	102	102	102
90	102	102	102
91	102	102	102
92	102	102	102
93	102	102	102
94	102	102	102
95	102	102	102
96	102	102	102
97	102	102	102
98	102	102	102
99	102	102	102
100	102	102	102
101	102	102	102
102	102	102	102
103	102	102	102
104	102	102	102
105	102	102	102
106	102	102	102
107	102	102	102
108	102	102	102
109	102	102	102
110	102	102	102
111	102	102	102
112	102	102	102
113	102	102	102
114	102	102	102
115	102	102	102
116	102	102	102
117	102	102	102
118	102	102	102
119	102	102	102
120	102	102	102
121	102	102	102
122	102	102	102
123	102	102	102
124	102	102	102
125	102	102	102
126	102	102	102
127	102	102	102
128	102	102	102
129	102	102	102
130	102	102	102
131	102	102	102
132	102	102	102
133	102	102	102
134	102	102	102
135	102	102	102
136	102	102	102
137	102	102	102
138	102	102	102
139	102	102	102
140	102	102	102
141	102	102	102
142	102	102	102
143	102	102	102
144	102	102	102
145	102	102	102
146	102	102	102
147	102	102	102
148	102	102	102
149	102	102	102
150	102	102	102
151	102	102	102
152	102	102	102
153	102	102	102
154	102	102	102
155	102	102	102
156	102	102	102
157	102	102	102
158	102	102	102
159	102	102	102
160	102	102	102
161	102	102	102
162	102	102	102
163	102	102	102
164	102	102	102
165	102	102	102
166	102	102	102
167	102	102	102
168	102	102	102
169	102	102	102
170	102	102	102
171	102	102	102
172	102	102	102
173	102	102	102
174	102	102	102
175	102	102	102
176	102	102	102
177	102	102	102
178	102	102	102
179	102	102	102
180	102	102	102
181	102	102	102
182	102	102	102
183	102	102	102
184	102	102	102
185	102	102	102
186	102	102	102
187	102	102	102
188	102	102	102
189	102	102	102
190	102	102	102
191	102	102	102
192	102	102	102
193	102	102	102
194	102	102	102
195	102	102	102
196	102	102	102
197	102	102	102
198	102	102	102
199	102	102	102
200	102	102	102

## 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
```

[illegible]

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;
```

```
D:\Querying> cd "C:\SQL\MS SQL Server\Binn" & cd "C:\SQL\MS SQL Server\Binn" --EXEC SQL 2
--EXEC SQL 2
SELECT d.dart_name, d.department_id, d.department_name, f.first_name, m.m_lname
FROM d.departments d, f.employees f, m.managers m
WHERE d.department_id = m.department_id AND f.department_id = d.department_id
END 
```

Resultset

	deptname	deptname	deptname	first_name	last_name
1	Accounting	Accounting	Accounting	John	King
2	Marketing	Marketing	Marketing	Neena	Kochhar
3	IT	IT	IT	Lex	De Haen
4	Finance	Finance	Finance	Michael	Stiles
5	Operations	Operations	Operations	Pat	Fay
6	Human Resources	Human Resources	Human Resources	Den	Raph
7	Legal	Legal	Legal	Timothy	Gaughan
8	Shipping	Shipping	Shipping	Erris	Cooper
9	Procurement	Procurement	Procurement	Yusef	Mudrapati
10	Contracting	Contracting	Contracting	Julia	Abel
11	Customer Support	Customer Support	Customer Support	Shelley	Stevens
12	Training	Training	Training	Sigal	Tobias
13	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
14	Manufacturing	Manufacturing	Manufacturing	James	Whalen
15	Product Development	Product Development	Product Development	Michael	Greenberg
16	Research and Development	Research and Development	Research and Development	Neena	Kochhar
17	Executive	Executive	Executive	John	King
18	Finance	Finance	Finance	Michael	Stiles
19	Operations	Operations	Operations	Pat	Fay
20	Human Resources	Human Resources	Human Resources	Den	Raph
21	Legal	Legal	Legal	Timothy	Gaughan
22	Shipping	Shipping	Shipping	Erris	Cooper
23	Procurement	Procurement	Procurement	Yusef	Mudrapati
24	Contracting	Contracting	Contracting	Julia	Abel
25	Customer Support	Customer Support	Customer Support	Shelley	Stevens
26	Training	Training	Training	Sigal	Tobias
27	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
28	Manufacturing	Manufacturing	Manufacturing	James	Whalen
29	Product Development	Product Development	Product Development	Michael	Greenberg
30	Research and Development	Research and Development	Research and Development	Neena	Kochhar
31	Executive	Executive	Executive	John	King
32	Finance	Finance	Finance	Michael	Stiles
33	Operations	Operations	Operations	Pat	Fay
34	Human Resources	Human Resources	Human Resources	Den	Raph
35	Legal	Legal	Legal	Timothy	Gaughan
36	Shipping	Shipping	Shipping	Erris	Cooper
37	Procurement	Procurement	Procurement	Yusef	Mudrapati
38	Contracting	Contracting	Contracting	Julia	Abel
39	Customer Support	Customer Support	Customer Support	Shelley	Stevens
40	Training	Training	Training	Sigal	Tobias
41	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
42	Manufacturing	Manufacturing	Manufacturing	James	Whalen
43	Product Development	Product Development	Product Development	Michael	Greenberg
44	Research and Development	Research and Development	Research and Development	Neena	Kochhar
45	Executive	Executive	Executive	John	King
46	Finance	Finance	Finance	Michael	Stiles
47	Operations	Operations	Operations	Pat	Fay
48	Human Resources	Human Resources	Human Resources	Den	Raph
49	Legal	Legal	Legal	Timothy	Gaughan
50	Shipping	Shipping	Shipping	Erris	Cooper
51	Procurement	Procurement	Procurement	Yusef	Mudrapati
52	Contracting	Contracting	Contracting	Julia	Abel
53	Customer Support	Customer Support	Customer Support	Shelley	Stevens
54	Training	Training	Training	Sigal	Tobias
55	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
56	Manufacturing	Manufacturing	Manufacturing	James	Whalen
57	Product Development	Product Development	Product Development	Michael	Greenberg
58	Research and Development	Research and Development	Research and Development	Neena	Kochhar
59	Executive	Executive	Executive	John	King
60	Finance	Finance	Finance	Michael	Stiles
61	Operations	Operations	Operations	Pat	Fay
62	Human Resources	Human Resources	Human Resources	Den	Raph
63	Legal	Legal	Legal	Timothy	Gaughan
64	Shipping	Shipping	Shipping	Erris	Cooper
65	Procurement	Procurement	Procurement	Yusef	Mudrapati
66	Contracting	Contracting	Contracting	Julia	Abel
67	Customer Support	Customer Support	Customer Support	Shelley	Stevens
68	Training	Training	Training	Sigal	Tobias
69	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
70	Manufacturing	Manufacturing	Manufacturing	James	Whalen
71	Product Development	Product Development	Product Development	Michael	Greenberg
72	Research and Development	Research and Development	Research and Development	Neena	Kochhar
73	Executive	Executive	Executive	John	King
74	Finance	Finance	Finance	Michael	Stiles
75	Operations	Operations	Operations	Pat	Fay
76	Human Resources	Human Resources	Human Resources	Den	Raph
77	Legal	Legal	Legal	Timothy	Gaughan
78	Shipping	Shipping	Shipping	Erris	Cooper
79	Procurement	Procurement	Procurement	Yusef	Mudrapati
80	Contracting	Contracting	Contracting	Julia	Abel
81	Customer Support	Customer Support	Customer Support	Shelley	Stevens
82	Training	Training	Training	Sigal	Tobias
83	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
84	Manufacturing	Manufacturing	Manufacturing	James	Whalen
85	Product Development	Product Development	Product Development	Michael	Greenberg
86	Research and Development	Research and Development	Research and Development	Neena	Kochhar
87	Executive	Executive	Executive	John	King
88	Finance	Finance	Finance	Michael	Stiles
89	Operations	Operations	Operations	Pat	Fay
90	Human Resources	Human Resources	Human Resources	Den	Raph
91	Legal	Legal	Legal	Timothy	Gaughan
92	Shipping	Shipping	Shipping	Erris	Cooper
93	Procurement	Procurement	Procurement	Yusef	Mudrapati
94	Contracting	Contracting	Contracting	Julia	Abel
95	Customer Support	Customer Support	Customer Support	Shelley	Stevens
96	Training	Training	Training	Sigal	Tobias
97	Quality Assurance	Quality Assurance	Quality Assurance	Ismael	Sciarra
98	Manufacturing	Manufacturing	Manufacturing	James	Whalen
99	Product Development	Product Development	Product Development	Michael	Greenberg
100	Research and Development	Research and Development	Research and Development	Neena	Kochhar

2. Create a query to connect to the DEPT00N (10.0.0.1) DEPT00N (10.0.0.1) Human Resources 10.0.0.1

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';
```

```

--DESCRIPTO D
select e.last_name , e.department_id, i.job_title, d.department_name , i.city
from employees e,
left join jobs i on
e.job_id = i.job_id
join departments d on
e.department_id = d.department_id
join locations l on
e.location_id = l.location_id
where i.city='Toronto';

```

id	name	department_id	job_id	department_name	city
1	De Haan	100	Manager	Toronto	
2	Patton	20	Marketing Representative	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)

```

```

--DESCRIPTO D
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)

```

ID_Empleado	Empleado	ID_Manager	Manager
1	De Haan	100	King
2	Patton	100	King
3	De Haan	100	King
4	Patton	100	King
5	Patton	100	King
6	Patton	100	King
7	Patton	100	King
8	Patton	100	King
9	Patton	100	King
10	Patton	100	King
11	Patton	100	King
12	Patton	100	King
13	Patton	100	King
14	Patton	100	King
15	Patton	100	King
16	Patton	100	King
17	Patton	100	King
18	Patton	100	King
19	Patton	100	King
20	Patton	100	King
21	Patton	100	King
22	Patton	100	King
23	Patton	100	King
24	Patton	100	King
25	Patton	100	King
26	Patton	100	King
27	Patton	100	King
28	Patton	100	King
29	Patton	100	King
30	Patton	100	King
31	Patton	100	King
32	Patton	100	King
33	Patton	100	King
34	Patton	100	King
35	Patton	100	King
36	Patton	100	King
37	Patton	100	King
38	Patton	100	King
39	Patton	100	King
40	Patton	100	King
41	Patton	100	King
42	Patton	100	King
43	Patton	100	King
44	Patton	100	King
45	Patton	100	King
46	Patton	100	King
47	Patton	100	King
48	Patton	100	King
49	Patton	100	King
50	Patton	100	King
51	Patton	100	King
52	Patton	100	King
53	Patton	100	King
54	Patton	100	King
55	Patton	100	King
56	Patton	100	King
57	Patton	100	King
58	Patton	100	King
59	Patton	100	King
60	Patton	100	King
61	Patton	100	King
62	Patton	100	King
63	Patton	100	King
64	Patton	100	King
65	Patton	100	King
66	Patton	100	King
67	Patton	100	King
68	Patton	100	King
69	Patton	100	King
70	Patton	100	King
71	Patton	100	King
72	Patton	100	King
73	Patton	100	King
74	Patton	100	King
75	Patton	100	King
76	Patton	100	King
77	Patton	100	King
78	Patton	100	King
79	Patton	100	King
80	Patton	100	King
81	Patton	100	King
82	Patton	100	King
83	Patton	100	King
84	Patton	100	King
85	Patton	100	King
86	Patton	100	King
87	Patton	100	King
88	Patton	100	King
89	Patton	100	King
90	Patton	100	King
91	Patton	100	King
92	Patton	100	King
93	Patton	100	King
94	Patton	100	King
95	Patton	100	King
96	Patton	100	King
97	Patton	100	King
98	Patton	100	King
99	Patton	100	King
100	Patton	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)

```

```

--DESCRIPTO D
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)

```

ID_Empleado	Empleado	ID_Manager	Manager
1	De Haan	100	King
2	Patton	100	King
3	De Haan	100	King
4	Patton	100	King
5	Patton	100	King
6	Patton	100	King
7	Patton	100	King
8	Patton	100	King
9	Patton	100	King
10	Patton	100	King
11	Patton	100	King
12	Patton	100	King
13	Patton	100	King
14	Patton	100	King
15	Patton	100	King
16	Patton	100	King
17	Patton	100	King
18	Patton	100	King
19	Patton	100	King
20	Patton	100	King
21	Patton	100	King
22	Patton	100	King
23	Patton	100	King
24	Patton	100	King
25	Patton	100	King
26	Patton	100	King
27	Patton	100	King
28	Patton	100	King
29	Patton	100	King
30	Patton	100	King
31	Patton	100	King
32	Patton	100	King
33	Patton	100	King
34	Patton	100	King
35	Patton	100	King
36	Patton	100	King
37	Patton	100	King
38	Patton	100	King
39	Patton	100	King
40	Patton	100	King
41	Patton	100	King
42	Patton	100	King
43	Patton	100	King
44	Patton	100	King
45	Patton	100	King
46	Patton	100	King
47	Patton	100	King
48	Patton	100	King
49	Patton	100	King
50	Patton	100	King
51	Patton	100	King
52	Patton	100	King
53	Patton	100	King
54	Patton	100	King
55	Patton	100	King
56	Patton	100	King
57	Patton	100	King
58	Patton	100	King
59	Patton	100	King
60	Patton	100	King
61	Patton	100	King
62	Patton	100	King
63	Patton	100	King
64	Patton	100	King
65	Patton	100	King
66	Patton	100	King
67	Patton	100	King
68	Patton	100	King
69	Patton	100	King
70	Patton	100	King
71	Patton	100	King
72	Patton	100	King
73	Patton	100	King
74	Patton	100	King
75	Patton	100	King
76	Patton	100	King
77	Patton	100	King
78	Patton	100	King
79	Patton	100	King
80	Patton	100	King
81	Patton	100	King
82	Patton	100	King
83	Patton	100	King
84	Patton	100	King
85	Patton	100	King
86	Patton	100	King
87	Patton	100	King
88	Patton	100	King
89	Patton	100	King
90	Patton	100	King
91	Patton	100	King
92	Patton	100	King
93	Patton	100	King
94	Patton	100	King
95	Patton	100	King
96	Patton	100	King
97	Patton	100	King
98	Patton	100	King
99	Patton	100	King
100	Patton	100	King

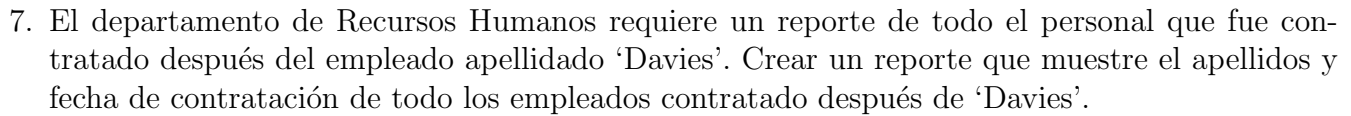
6. Crear un reporte que muestre los No de Departamento y Apellidos de todos los empleados, asimismo adicionar una columna con los Apellidos de todos empleados que trabajan en el mismo departamento. Etiquetar esta columna como Colega.

```

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;

```



[illegible]

- ```
-- Ejercicio 8
select s.last_name "EMPLOYEE", s.hire_date "HIRE_DATE", s.job_title "JOB_TITLE",
s.department_id "DEPARTMENT"
from employees s
join emp_dept d
on s.employee_id = d.employee_id
-- s.hire_date < s.hire_date
order by s.hire_date;
```

## 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,
```

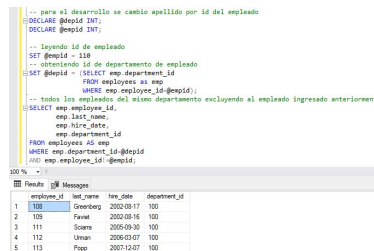
```
emp.hire_date,
```

```
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 @empid INT;
--DECLARE @depid 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         | Gilesberg | 2002-08-17 | 100           |
| 109         | Favari    | 2002-08-16 | 100           |
| 111         | Scarna    | 2005-09-30 | 100           |
| 112         | Uman      | 2005-02-07 | 100           |
| 113         | Prip      | 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         | Solara    | 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 contienen "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         | Rasmussen | 50            |
| 115         | Rhoo      | 30            |
| 116         | Sales     | 30            |
| 117         | Tobias    | 30            |
| 118         | Hunns     | 30            |
| 119         | Cohen     | 30            |
| 120         | Taylor    | 50            |
| 121         | Rodur     | 50            |
| 122         | Sullivan  | 50            |
| 123         | Geevi     | 50            |

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 apellide '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\_id=manking.employee\_id) AS depking  
 ON emp.department\_id=depking.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';
```

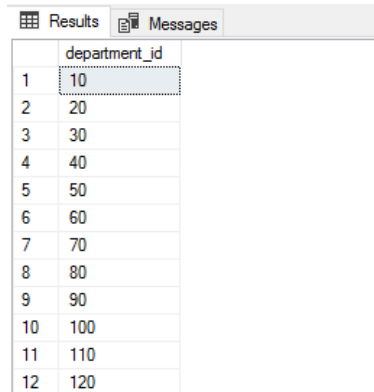
| department_id | last_name | job_title                     |
|---------------|-----------|-------------------------------|
| 90            | King      | President                     |
| 90            | Kochhar   | Administration Vice President |
| 90            | De Haan   | Administration Vice President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

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';
```

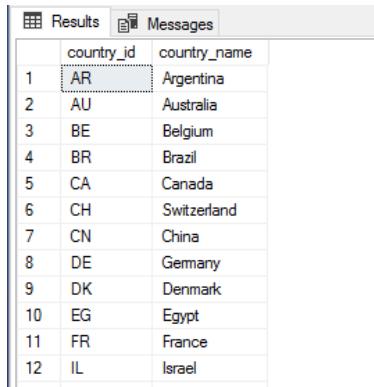


The screenshot shows a database query result with a table titled 'Results'. The table has two columns: 'department\_id' and an unlabeled column. The data is as follows:

|    | 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 a table titled 'Results'. The table has two columns: 'country\_id' and 'country\_name'. The data is as follows:

|    | 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);
```

| Results |         | Messages      |
|---------|---------|---------------|
|         | 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;
```

| Results |             | Messages |
|---------|-------------|----------|
|         | 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;
```

| Results   |               | Messages         |
|-----------|---------------|------------------|
| 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         |