

UNERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

INFORME DE LABORATORIO No 01

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Cuadros Quiroga

Integrantes:

Jhony Mamani Limache	(2013046566)
Colque Ticona Carlos	(2013046500)
Luis Zavala Venegas	(2010037899)
Moreno Cáceres Renzo	(2013047246)
Ronald Ordoñez Quilli	(2015052821)
Condori Tito Hernan	(2009034553)

Índice

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_ACCOUNT	9000.00
11	Chen	FI_ACCOUNT	8200.00
12	Sciarra	FI_ACCOUNT	7700.00

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

job_id	job_title	min_salary	max_salary
AC_ACCOUNT	Public Accountant	4200	9000
AC_MGR	Accounting Manager	8200	16000
AD_ASST	Administration Assistant	3000	6000
AD_PRES	President	20080	40000
AD_VP	Administration Vice President	15000	30000
FI_ACCOUNT	Accountant	4200	9000
FI_MGR	Finance Manager	8200	16000
HR_REP	Human Resources Representative	4000	9000
IT_PROG	Programmer	4000	10000
MARK_MGR	Marketing Manager	9000	15000

- 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
100	King	288000.00
101	Kochhar	204000.00
102	De Haan	204000.00
103	Hunold	108000.00
104	Ernst	72000.00
105	Austin	57600.00
106	Pataballa	57600.00
107	Lorentz	50400.00
108	Greenberg	144096.00
109	Faviet	108000.00

2. Actividad No 02 – Reconociendo la estructura

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

SP_HELP 'DEPARTMENTS'

-- 1
SP_HELP departments,]

100 %

Results
Messages

	Name	Owner	Type	Created_datetime
1	departments	dbo	user table	2018-04-21 14:47:27.640

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

	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

	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	PRIMARY KEY (clustered)	dept_id_pk	(n/a)	(n/a)	(n/a)	(n/a)	department_id
2	FOREIGN KEY	dept_loc_fk	No Action	No Action	Enabled	Is_For_Replication	location_id
3							REFERENCE...

	Table is referenced by foreign key
1	Human_Resources.dbo.employees: emp_dept_fk
2	Human_Resources.dbo.job_history: jhist_dept_fk

2. El departamento de Recursos Humanos requiere un reporte que muestre los campos: employee_id, last_name y job_id, asi como 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

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;
```

```
-- //////////////////////////////////////
-- 3
[SELECT DISTINCT job_id FROM employees;
```

100 %

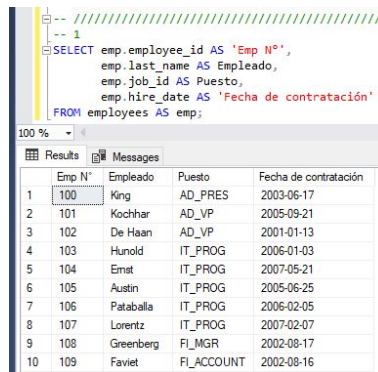
Results Messages

	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

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 text:

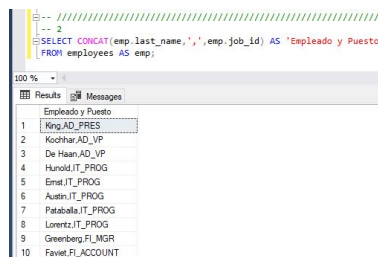
```
-- 1
SELECT emp.employee_id AS 'Emp N',
emp.last_name AS Empleado,
emp.job_id AS Puesto,
emp.hire_date AS 'Fecha de contratación'
FROM employees AS emp;
```

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 10 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

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 text:

```
-- 2
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 10 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
Faviet,FI_ACCOUNT

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;

```

The screenshot shows a SQL IDE with a query window and a results window. The query window contains the following SQL code:

```

-- 3
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;

```

The results window shows a table with 10 rows and 1 column. The column header is 'Los empleados'. The rows contain concatenated strings of employee data.

	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_...

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
1	Livingston	80
2	Grant	NULL
3	Johnson	80
4	Taylor	50
5	Fleur	50
6	Sullivan	50
7	Geoni	50
8	Sarchand	50
9	Bull	50
10	Dellinger	50
11	Cabrio	50
12	Chung	50
13	Dilly	50
14	Gates	50
15	Perkins	50
16	Bell	50
17	Everett	50
18	McCain	50
19	Jones	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;
```

	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	Uman	FI_ACCOUNT	7800.00
7	Popp	FI_ACCOUNT	6900.00
8	Raphaely	PU_MAN	11000.00
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	Cambraut	SA_MAN	11000.00
15	Zlotkey	SA_MAN	10500.00
16	Tucker	SA_REP	10000.00
17	Bernstein	SA_REP	9500.00
18	Hall	SA_REP	9000.00
19	Olsen	SA_REP	8000.00

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;
```

	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;
```

	last_name	department_id
1	Atkinson	50
2	Bell	50
3	Bissot	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
13	Fleaur	50
14	Fripp	50
15	Gates	50
16	Gee	50
17	Geoni	50
18	Grant	50
19	Hartstein	20

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);
```

	Empleado	Salario Mensual
1	Weiss	8000.00
2	Fripp	8200.00
3	Kaufling	7900.00
4	Vollman	6500.00
5	Mourgos	5800.00
6	Fay	6000.00

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';
```

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;
```

	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	Erasmus	12000.00	0.30
10	Ozer	11500.00	0.25
11	Cambraut	11000.00	0.30
12	Abel	11000.00	0.30
13	Raphaely	11000.00	NULL
14	Vishney	10500.00	0.25
15	Zlotkey	10500.00	0.20
16	King	10000.00	0.35
17	Tucker	10000.00	0.30
18	Bloom	10000.00	0.20
19	Baer	10000.00	NULL

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

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
```

	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.

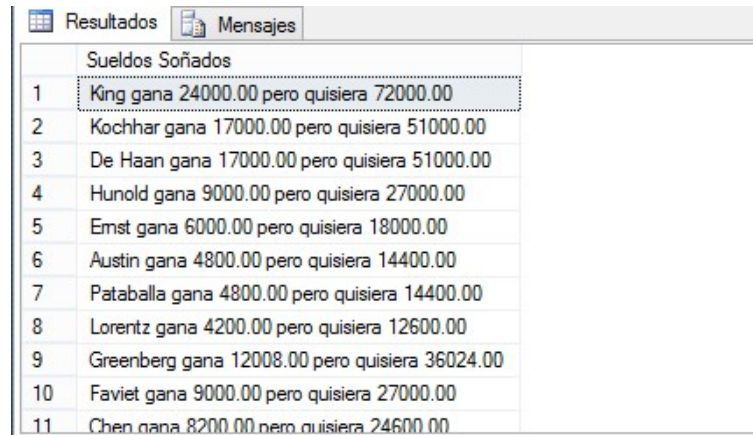
```
select last_name 'Empleado',salary 'Salario Mensual',commission_pct from employees where salary >5000 and salary <12000 and (department_id = 20 or department_id = 50) and commission_pct = 0.20;  
go
```

Empleado	Salario Mensual	commission_pct

5. Actividad No 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	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
```


Resultados		Mensajes
	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
```

Resultados		Mensajes	
	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

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

```

Resultados		Mensajes
	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

```

Resultados		Mensajes	
	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	Greenberg	Finance Manager	0
10	Faviet	Accountant	0
11	Chen	Accountant	0

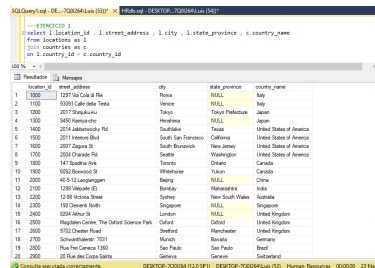
6. Actividad No 07 – Funciones de Agrupación

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.
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.
4. Determinar el número de Administradores o Supervisores utilizar la columna `manager_id` para esto. Etiquetar la columna como No de Administradores
5. Encontrar la diferencia entre el máximo y mínimo salario de los empleados. Etiquetar la columna como Diferencia
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.
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.

7. 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
```



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

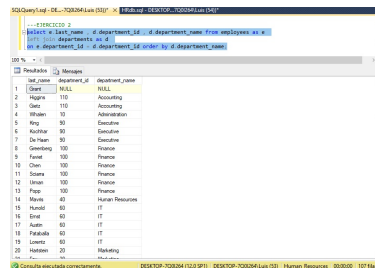
```
--169600200 1
select l.location_id , l.street_address , l.city , l.state_province , c.country_name
from locations as l
join countries as c
on l.country_id = c.country_id
```

The results grid displays the following data:

location_id	street_address	city	state_province	country_name
1	1237 Via Cola di Rie	Roma	NULL	Italy
2	1000 10000 Lakeside Drive	Venice	NULL	Italy
3	1000 10000 Lakeside Drive	Tokyo	Tokyo Prefecture	Japan
4	1000 10000 Lakeside Drive	Hongkong	NULL	China
5	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
6	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
7	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
8	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
9	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
10	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
11	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
12	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
13	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
14	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
15	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
16	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
17	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
18	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
19	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa
20	1000 10000 Lakeside Drive	South Africa	Western Cape	South Africa

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;
```



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

```
--169600200 2
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;
```

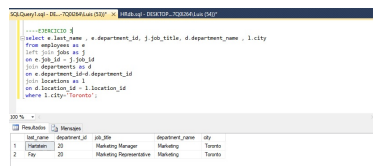
The results grid displays the following data:

last_name	department_id	department_name
Chen	100	Accounting
Higgins	100	Accounting
Baer	100	Accounting
Whalen	100	Accounting
King	100	Accounting
De Haan	100	Accounting
Greenberg	100	Accounting
Ford	100	Accounting
Chen	100	Accounting
Scooter	100	Accounting
Ullrich	100	Accounting
Peter	100	Accounting
Blake	100	Accounting
Scott	100	Accounting
Timothy	100	Accounting
Ernst	100	Accounting
David	100	Accounting
Neena	100	Accounting
Lex	100	Accounting
De Haan	100	Accounting
Chen	100	Accounting

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';

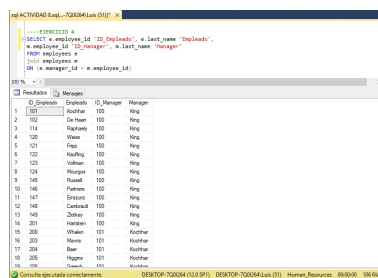


Query: SELECT * FROM employees e, departments d, locations l WHERE d.location_id = l.location_id AND l.city = 'Toronto';

emp_id	emp_name	job_id	department_id	department_name	location_id	city
1	Steven	ADP	90	Marketing	1400	Toronto
2	Neena	ANL	90	Marketing	1400	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)

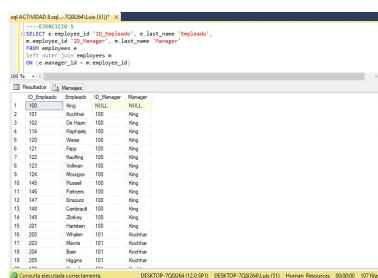


Query: SELECT e.employee_id 'ID_Empleado', e.last_name 'Empleado', m.employee_id 'ID_Manager', m.last_name 'Manager' FROM employees e join employees m ON (e.manager_id = m.employee_id);

ID_Empleado	Empleado	ID_Manager	Manager
100	Koussair	100	King
101	De Haan	100	King
102	Patton	100	King
103	Alton	100	King
104	Chen	100	King
105	Bradley	100	King
106	Abel	100	King
107	Smith	100	King
108	Popper	100	King
109	Turner	100	King
110	Russell	100	King
111	Patton	100	King
112	De Haan	100	King
113	Patton	100	King
114	Patton	100	King
115	Patton	100	King
116	Patton	100	King
117	Patton	100	King
118	Patton	100	King
119	Patton	100	King
120	Patton	100	King
121	Patton	100	King
122	Patton	100	King
123	Patton	100	King
124	Patton	100	King
125	Patton	100	King
126	Patton	100	King
127	Patton	100	King
128	Patton	100	King
129	Patton	100	King
130	Patton	100	King
131	Patton	100	King
132	Patton	100	King
133	Patton	100	King
134	Patton	100	King
135	Patton	100	King
136	Patton	100	King
137	Patton	100	King
138	Patton	100	King
139	Patton	100	King
140	Patton	100	King
141	Patton	100	King
142	Patton	100	King
143	Patton	100	King
144	Patton	100	King
145	Patton	100	King
146	Patton	100	King
147	Patton	100	King
148	Patton	100	King
149	Patton	100	King
150	Patton	100	King
151	Patton	100	King
152	Patton	100	King
153	Patton	100	King
154	Patton	100	King
155	Patton	100	King
156	Patton	100	King
157	Patton	100	King
158	Patton	100	King
159	Patton	100	King
160	Patton	100	King
161	Patton	100	King
162	Patton	100	King
163	Patton	100	King
164	Patton	100	King
165	Patton	100	King
166	Patton	100	King
167	Patton	100	King
168	Patton	100	King
169	Patton	100	King
170	Patton	100	King
171	Patton	100	King
172	Patton	100	King
173	Patton	100	King
174	Patton	100	King
175	Patton	100	King
176	Patton	100	King
177	Patton	100	King
178	Patton	100	King
179	Patton	100	King
180	Patton	100	King
181	Patton	100	King
182	Patton	100	King
183	Patton	100	King
184	Patton	100	King
185	Patton	100	King
186	Patton	100	King
187	Patton	100	King
188	Patton	100	King
189	Patton	100	King
190	Patton	100	King
191	Patton	100	King
192	Patton	100	King
193	Patton	100	King
194	Patton	100	King
195	Patton	100	King
196	Patton	100	King
197	Patton	100	King
198	Patton	100	King
199	Patton	100	King
200	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)



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);

ID_Empleado	Empleado	ID_Manager	Manager
100	Koussair	100	King
101	De Haan	100	King
102	Patton	100	King
103	Alton	100	King
104	Chen	100	King
105	Bradley	100	King
106	Abel	100	King
107	Smith	100	King
108	Popper	100	King
109	Turner	100	King
110	Russell	100	King
111	Patton	100	King
112	De Haan	100	King
113	Patton	100	King
114	Patton	100	King
115	Patton	100	King
116	Patton	100	King
117	Patton	100	King
118	Patton	100	King
119	Patton	100	King
120	Patton	100	King
121	Patton	100	King
122	Patton	100	King
123	Patton	100	King
124	Patton	100	King
125	Patton	100	King
126	Patton	100	King
127	Patton	100	King
128	Patton	100	King
129	Patton	100	King
130	Patton	100	King
131	Patton	100	King
132	Patton	100	King
133	Patton	100	King
134	Patton	100	King
135	Patton	100	King
136	Patton	100	King
137	Patton	100	King
138	Patton	100	King
139	Patton	100	King
140	Patton	100	King
141	Patton	100	King
142	Patton	100	King
143	Patton	100	King
144	Patton	100	King
145	Patton	100	King
146	Patton	100	King
147	Patton	100	King
148	Patton	100	King
149	Patton	100	King
150	Patton	100	King
151	Patton	100	King
152	Patton	100	King
153	Patton	100	King
154	Patton	100	King
155	Patton	100	King
156	Patton	100	King
157	Patton	100	King
158	Patton	100	King
159	Patton	100	King
160	Patton	100	King
161	Patton	100	King
162	Patton	100	King
163	Patton	100	King
164	Patton	100	King
165	Patton	100	King
166	Patton	100	King
167	Patton	100	King
168	Patton	100	King
169	Patton	100	King
170	Patton	100	King
171	Patton	100	King
172	Patton	100	King
173	Patton	100	King
174	Patton	100	King
175	Patton	100	King
176	Patton	100	King
177	Patton	100	King
178	Patton	100	King
179	Patton	100	King
180	Patton	100	King
181	Patton	100	King
182	Patton	100	King
183	Patton	100	King
184	Patton	100	King
185	Patton	100	King
186	Patton	100	King
187	Patton	100	King
188	Patton	100	King
189	Patton	100	King
190	Patton	100	King
191	Patton	100	King
192	Patton	100	King
193	Patton	100	King
194	Patton	100	King
195	Patton	100	King
196	Patton	100	King
197	Patton	100	King
198	Patton	100	King
199	Patton	100	King
200	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

```
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]

- ```
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;
```

```

--Exercise 7
SELECT h.hire_name, e.last_name, e.hire_date
FROM employees e
JOIN employees emp
ON (emp.hire_name = 'Trevor')
WHERE emp.hire_date = h.hire_date

```

10 %

| emp_name    | last_name | hire_date  |
|-------------|-----------|------------|
| Paul        | Tutor     | 2005-03-08 |
| David       | Burnton   | 2005-03-08 |
| Paul        | Val       | 2005-03-08 |
| Christopher | Chen      | 2005-03-08 |
| Wendy       | Chantel   | 2005-10-09 |
| David       | Tyebk     | 2007-11-09 |
| Jeffrey     | Smith     | 2005-12-09 |
| Joan        | Deane     | 2005-12-10 |
| Scott       | Stark     | 2005-11-09 |
| Olga        | Wintern   | 2007-11-11 |
| Georgie     | Turner    | 2005-12-09 |
| Mattias     | Munoz     | 2005-03-08 |
| David       | Lee       | 2005-03-08 |
| Yusef       | Ali       | 2005-03-08 |
| David       | Bards     | 2005-03-08 |
| David       | Law       | 2005-03-08 |
| Heather     | Burn      | 2005-03-08 |
| Yusef       | Ali       | 2005-03-08 |

- ```
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;
```


23

8. 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,
```

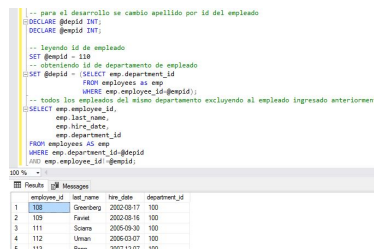
```
emp.hire_date,
```

```
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	Scars	2005-09-30	100
112	Unam	2006-03-07	100
113	Papa	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	Rachney	30
115	Abene	30
116	Bede	30
117	Tobias	30
118	Herman	30
119	Colmenares	30
120	Taylor	30
121	Russur	30
122	Sullivan	30
123	Gentry	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 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_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';

```

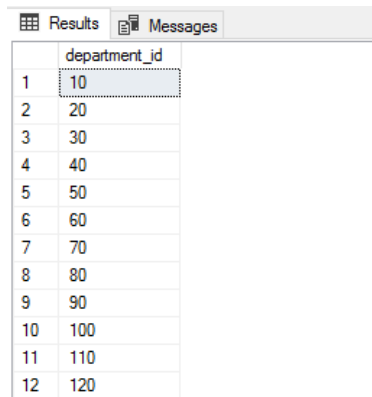
department_id	last_name	job_title
90	King	President
90	Kochhar	Administration Vice President
90	De Haan	Administration Vice President

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.

9. 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 interface with a 'Results' tab. The query result is a table with two columns: an index from 1 to 12 and 'department_id'. The values in the 'department_id' column are 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, and 120. The first row, containing the value 10, 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 interface with a 'Results' tab. The query result is a table with two columns: 'country_id' and 'country_name'. The rows list countries with their IDs: AR (Argentina), AU (Australia), BE (Belgium), BR (Brazil), CA (Canada), CH (Switzerland), CN (China), DE (Germany), DK (Denmark), EG (Egypt), FR (France), and IL (Israel). The first row, containing 'AR' and 'Argentina', 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, depart-
ment_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