

M5_AE2_ABP-Ejercicio individual [Actividad Evaluada]

M5_AE2_ABP_Ejercicio individual [Actividad Evaluada]

Ejercicio individual

Contexto

Para este ejercicio deberás de crear una tabla con los siguientes atributos.

+-----+	
	empleados
+-----+	
	id (PK)
	nombre
	departamento
	salario
+-----+	

Ahora ingresa los siguientes registros para poder hacer consultas:

```
INSERT INTO empleados (id, nombre, departamento, salario) VALUES
(1, 'Ana García', 'Recursos Humanos', 32000),
(2, 'Luis Pérez', 'Marketing', 35000),
(3, 'Carlos Díaz', 'Ventas', 27000),
(4, 'María López', 'Contabilidad', 40000),
(5, 'Pedro Martínez', 'Desarrollo', 45000),
(6, 'Julia Fernández', 'Recursos Humanos', 31000),
(7, 'Juan Rodríguez', 'Marketing', 38000),
(8, 'Elena Sánchez', 'Ventas', 26000),
(9, 'David González', 'Contabilidad', 42000),
(10, 'Raquel Pérez', 'Desarrollo', 46000),
(11, 'Fernando García', 'Recursos Humanos', 33000),
(12, 'Isabel Ruiz', 'Marketing', 36000),
(13, 'Sergio Gómez', 'Ventas', 28000),
(14, 'Carmen Romero', 'Contabilidad', 39000),
(15, 'José Torres', 'Desarrollo', 48000);
```

Crea las consultas SELECT que cumplan con las siguientes descripciones:

- Actualiza el salario del empleado con ID=3 a 30000
- Proyecta todos los empleados que trabajen en el departamento de 'Ventas'.
- Calcula el salario promedio para cada departamento.
- Proyecta solo los nombres de todos los departamentos.
- Proyecta los dos empleados con los salarios más bajos, ordenados de forma ascendente por salario.
- Entrega un archivo .sql que contenga las 5 consultas que ejecutaste en el entorno de trabajo.

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar
Última modificación	-

DESARROLLO M5_AE2_ABP-EJERCICIO INDIVIDUAL

```
/*
-- Crear tabla empleados con los siguientes atributos
*/
CREATE TABLE empleados (
    id INTEGER PRIMARY KEY,
    nombre TEXT NOT NULL,
    departamento TEXT NOT NULL,
    salario REAL NOT NULL
);

/*
-- Insertar registros
*/
INSERT INTO empleados (id, nombre, departamento, salario) VALUES
(1, 'Ana García', 'Recursos Humanos', 32000),
(2, 'Luis Pérez', 'Marketing', 35000),
(3, 'Carlos Díaz', 'Ventas', 27000),
(4, 'María López', 'Contabilidad', 40000),
(5, 'Pedro Martínez', 'Desarrollo', 45000),
(6, 'Julia Fernández', 'Recursos Humanos', 31000),
(7, 'Juan Rodríguez', 'Marketing', 38000),
(8, 'Elena Sánchez', 'Ventas', 26000),
(9, 'David González', 'Contabilidad', 42000),
(10, 'Raquel Pérez', 'Desarrollo', 46000),
(11, 'Fernando García', 'Recursos Humanos', 33000),
(12, 'Isabel Ruiz', 'Marketing', 36000),
(13, 'Sergio Gómez', 'Ventas', 28000),
(14, 'Carmen Romero', 'Contabilidad', 39000),
(15, 'José Torres', 'Desarrollo', 48000);

/*
-- 1. Actualiza el salario del empleado con ID=3 a 30000
*/
UPDATE empleados
SET salario = 30000
WHERE id = 3;
```

```
/*
-- 1. Actualiza el salario del empleado con ID=3 a 30000
*/
UPDATE empleados
SET salario = 30000
WHERE id = 3;

-- select * from empleados where Id=3

/*
-- 2. Proyecta todos los empleados que trabajen en el departamento de
*/
SELECT *
FROM empleados
```

empleados 1 X

select * from empleados where Id=3 | Enter a SQL expression to filter results (use Ctrl+Space)

123 id	AZ nombre	AZ departamento	123 salario
3	Carlos Díaz	Ventas	30.000

```

/*
-- 2. Proyecta todos los empleados que trabajen en el departamento de 'Ventas'
*/

```

```

SELECT *
FROM empleados
WHERE departamento = 'Ventas';

```

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

```

/*
-- 2. Proyecta todos los empleados que trabajen en el departamento de 'Ventas'
*/
SELECT *
FROM empleados
WHERE departamento = 'Ventas';

```

The results grid, titled "empleados 1", displays the following data:

	id	nombre	departamento	salario
1	8	Elena Sánchez	Ventas	26.000
2	13	Sergio Gómez	Ventas	28.000
3	3	Carlos Díaz	Ventas	30.000

```

/*
-- 3. Calcula el salario promedio para cada departamento
*/

```

```

SELECT departamento, ROUND(AVG(salario),2) AS salario_promedio
FROM empleados
GROUP BY departamento;

```

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

```

/*
-- 3. Calcula el salario promedio para cada departamento
*/
SELECT departamento, ROUND(AVG(salario),2) AS salario_promedio
FROM empleados
GROUP BY departamento;

```

The results grid, titled "empleados 1", displays the following data:

	departamento	salario_promedio
1	Marketing	36.333,33
2	Desarrollo	46.333,33
3	Contabilidad	40.333,33
4	Ventas	28.000
5	Recursos Humanos	32.000

```

/*
-- 4. Proyecta solo los nombres de todos los departamentos (sin repetir)
*/
SELECT DISTINCT departamento
FROM empleados;

```

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

```

/*
-- 4. Proyecta solo los nombres de todos los departamentos (sin repetir)
*/
SELECT DISTINCT departamento
FROM empleados;

```

The results pane shows the output of the query, displaying a table with 5 rows and 1 column (departamento). The results are sorted alphabetically (A-Z).

	A-Z departamento
1	Marketing
2	Desarrollo
3	Contabilidad
4	Ventas
5	Recursos Humanos

```

/*
-- 5. Proyecta los dos empleados con los salarios más bajos, ordenados de forma
ascendente por salario
*/
SELECT *
FROM empleados
ORDER BY salario ASC
LIMIT 2;

```

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

```

/*
-- 5. Proyecta los dos empleados con los salarios más bajos, ordenados de forma asc
*/
SELECT *
FROM empleados
ORDER BY salario ASC
LIMIT 2;

```

The results pane shows the output of the query, displaying a table with 5 columns (id, nombre, departamento, salario) and 2 rows. The results are sorted by salary in ascending order.

	123 id	A-Z nombre	A-Z departamento	123 salario
1	8	Elena Sánchez	Ventas	26.000
2	13	Sergio Gómez	Ventas	28.000