**1. Consulta con GROUP BY: Escribe una consulta que muestre la cantidad de usuarios por edad, agrupados por rangos (por ejemplo, 18-25, 26-35, etc.) en la tabla "Usuarios".**

SELECT

CASE

WHEN Edad BETWEEN 18 AND 25 THEN '18-25'

WHEN Edad BETWEEN 26 AND 35 THEN '26-35'

WHEN Edad BETWEEN 36 AND 45 THEN '36-45'

ELSE '46+'

END AS RangoEdad,

COUNT(\*) AS CantidadAspirantes

FROM Aspirantes

GROUP BY

CASE

WHEN Edad BETWEEN 18 AND 25 THEN '18-25'

WHEN Edad BETWEEN 26 AND 35 THEN '26-35'

WHEN Edad BETWEEN 36 AND 45 THEN '36-45'

ELSE '46+'

END;

Esta consulta utiliza la cláusula CASE para asignar cada edad a un rango específico. Luego, utiliza la función de agregación COUNT(\*) para contar la cantidad de aspirantes en cada rango. La consulta agrupa los resultados por el rango de edad y muestra la cantidad de aspirantes en cada uno.

Esta consulta repetirá la expresión CASE en la cláusula GROUP BY, asegurando que se pueda agrupar correctamente por el rango de edad.

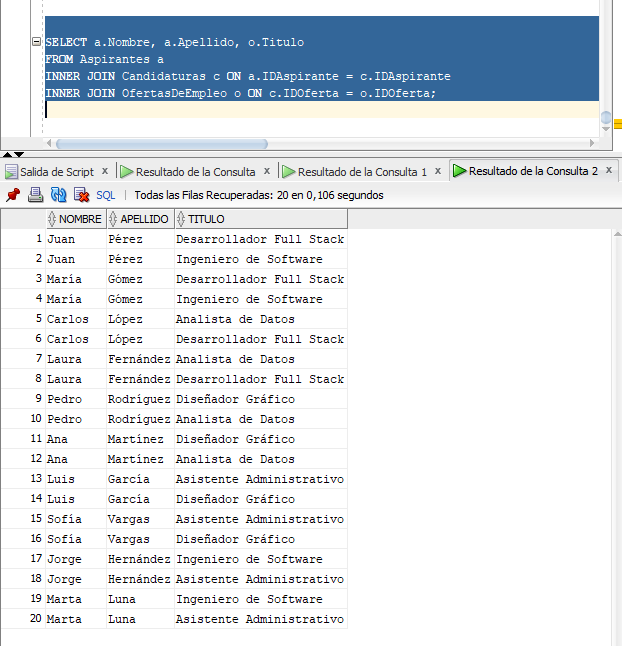
**2. Consulta con INNER JOIN: Realiza una consulta que muestre el nombre de los usuarios y los títulos de los empleos a los que se han inscrito. Utiliza las tablas "Usuarios" y "Inscripciones" en la consulta.**

SELECT a.Nombre, a.Apellido, o.Titulo

FROM Aspirantes a

INNER JOIN Candidaturas c ON a.IDAspirante = c.IDAspirante

INNER JOIN OfertasDeEmpleo o ON c.IDOferta = o.IDOferta;



Esta consulta realiza un **INNER JOIN** entre las tablas "Aspirantes" y "Candidaturas" usando la columna **IDAspirante**, y luego realiza otro **INNER JOIN** entre las tablas "Candidaturas" y "OfertasDeEmpleo" usando la columna **IDOferta**. De esta manera, se relacionan los aspirantes con las candidaturas y las candidaturas con las ofertas de empleo.

La consulta selecciona el nombre y apellido de los aspirantes de la tabla "Aspirantes" y el título de las ofertas de la tabla "OfertasDeEmpleo" para las inscripciones correspondientes en la tabla "Candidaturas".

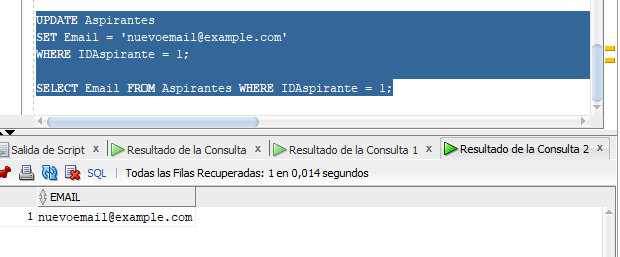
**3. Actualización con UPDATE: Actualiza el campo "Email" del usuario con ID = 1 en la tabla "Usuarios" y cambia su valor a "nuevoemail@example.com".**

UPDATE Aspirantes

SET Email = 'nuevoemail@example.com'

WHERE IDAspirante = 1;

SELECT Email FROM Aspirantes WHERE IDAspirante = 1;

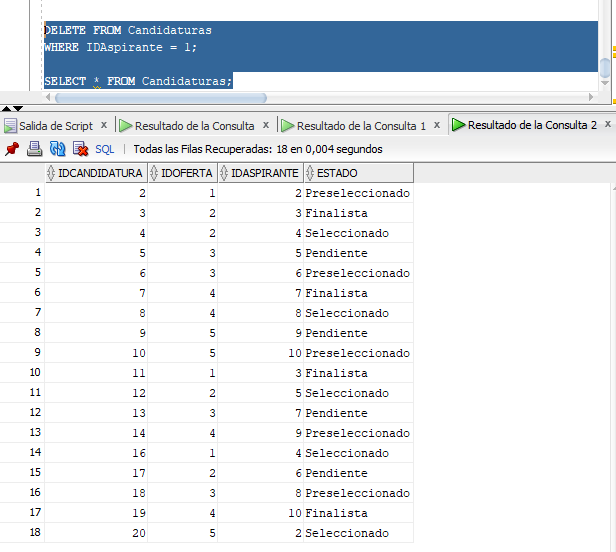


Esta consulta utiliza la sentencia **UPDATE** para modificar el campo "Email" en la tabla "Aspirantes". La cláusula **SET** establece el nuevo valor del campo y la cláusula **WHERE** especifica la condición para seleccionar el aspirante con IDAspirante = 1.

**4. Eliminación con DELETE: Elimina todas las inscripciones de la tabla "Inscripciones" que correspondan al usuario con ID = 1.**

DELETE FROM Candidaturas

WHERE IDAspirante = 1;



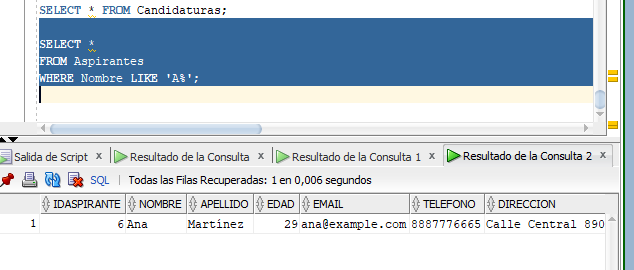
Esta consulta utiliza la sentencia **DELETE** para eliminar filas de la tabla "Candidaturas". La cláusula **WHERE** especifica la condición para seleccionar las inscripciones que corresponden al usuario con IDAspirante = 1.

**5. Consulta con WHERE: Escribe una consulta que muestre los usuarios cuyo nombre empiece con la letra "A" en la tabla "Usuarios".**

SELECT \*

FROM Aspirantes

WHERE Nombre LIKE 'A%';

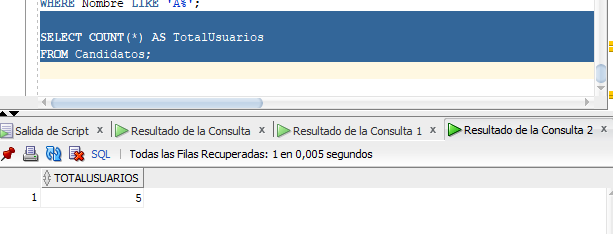


Esta consulta selecciona todas las columnas de la tabla "Aspirantes" donde el valor del campo "Nombre" comienza con la letra "A" seguida de cualquier número de caracteres. El símbolo "%" se utiliza como comodín para representar cualquier secuencia de caracteres.

**6. Consulta con COUNT: Realiza una consulta que muestre el número total de usuarios en la tabla "Usuarios".**

SELECT COUNT(\*) AS TotalUsuarios

FROM Candidatos;



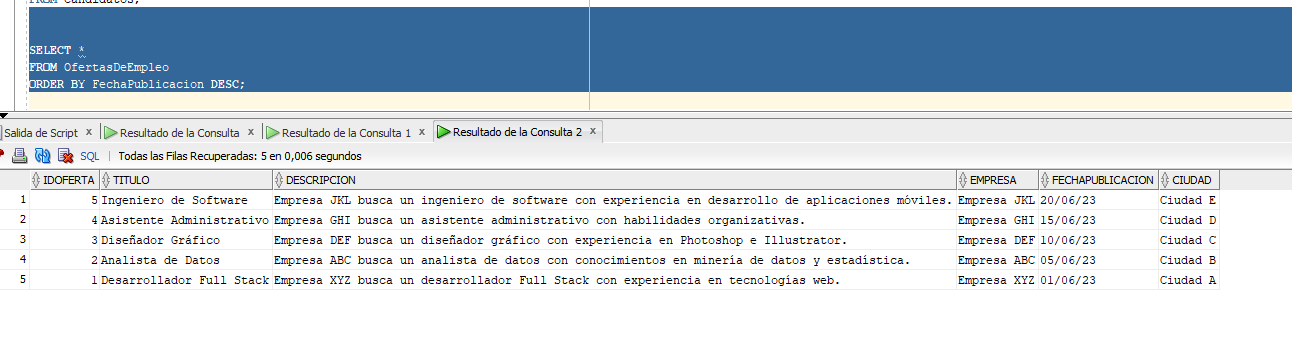
Esta consulta utiliza la función COUNT(\*) para contar todas las filas de la tabla "Candidatos". El alias "TotalUsuarios" se utiliza para dar nombre al resultado obtenido.

**7. Consulta con ORDER BY: Muestra los empleos de la tabla "Empleos" ordenados por fecha de publicación de forma descendente.**

SELECT \*

FROM OfertasDeEmpleo

ORDER BY FechaPublicacion DESC;



Esta consulta selecciona todas las columnas de la tabla "OfertasDeEmpleo" y las ordena en función de la columna "FechaPublicacion" en orden descendente, lo que mostrará los empleos más recientes primero.

**8. Consulta con HAVING: Escribe una consulta que muestre los rangos de edad y el número de usuarios por rango, pero solo para aquellos rangos que tengan más de 5 usuarios. Utiliza la tabla "Usuarios".**

SELECT

CASE

WHEN Edad BETWEEN 18 AND 25 THEN '18-25'

WHEN Edad BETWEEN 26 AND 35 THEN '26-35'

WHEN Edad BETWEEN 36 AND 45 THEN '36-45'

ELSE '46+'

END AS RangoEdad,

COUNT(\*) AS NumeroAspirantes

FROM Aspirantes

GROUP BY

CASE

WHEN Edad BETWEEN 18 AND 25 THEN '18-25'

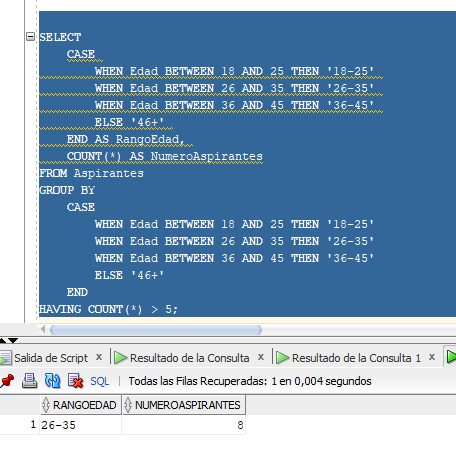
WHEN Edad BETWEEN 26 AND 35 THEN '26-35'

WHEN Edad BETWEEN 36 AND 45 THEN '36-45'

ELSE '46+'

END

HAVING COUNT(\*) > 5;



**9. Consulta con INNER JOIN y WHERE: Muestra los nombres de los usuarios y los títulos de los empleos a los que se han inscrito, pero solo para aquellos empleos que estén ubicados en una ciudad específica. Utiliza las tablas "Usuarios", "Inscripciones" y "Empleos".**

SELECT

A.Nombre,

A.Apellido,

O.Titulo

FROM

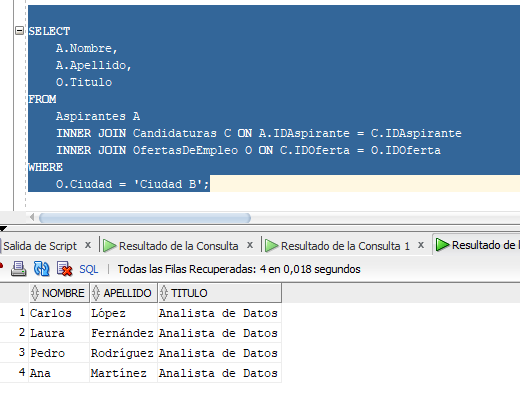
Aspirantes A

INNER JOIN Candidaturas C ON A.IDAspirante = C.IDAspirante

INNER JOIN OfertasDeEmpleo O ON C.IDOferta = O.IDOferta

WHERE

O.Ciudad = 'Ciudad B';



En esta consulta, se realiza un **INNER JOIN** entre las tablas "Aspirantes", "Candidaturas" y "OfertasDeEmpleo" utilizando las claves primarias y foráneas correspondientes. Luego, se utiliza la cláusula **WHERE** para filtrar solo los empleos que estén ubicados en la ciudad específica que deseas.

**10. Consulta con GROUP BY y COUNT: Escribe una consulta que muestre los títulos de los empleos y la cantidad de usuarios inscritos en cada uno, ordenados de mayor a menor cantidad de inscripciones. Utiliza las tablas "Empleos" e "Inscripciones"**

SELECT

O.Titulo,

COUNT(C.IDAspirante) AS CantidadAspirantes

FROM

OfertasDeEmpleo O

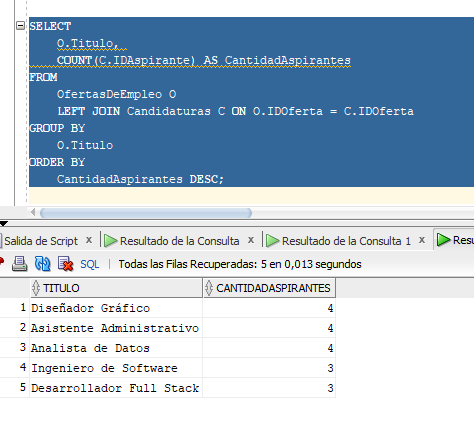
LEFT JOIN Candidaturas C ON O.IDOferta = C.IDOferta

GROUP BY

O.Titulo

ORDER BY

CantidadAspirantes DESC;



En esta consulta, se utiliza un **LEFT JOIN** entre las tablas "OfertasDeEmpleo" y "Candidaturas" para obtener todas las ofertas de empleo, incluso si no tienen candidaturas asociadas. Luego, se utiliza la función de agregación **COUNT** para contar el número de aspirantes inscritos en cada oferta. La cláusula **GROUP BY** agrupa los resultados por el título de la oferta. Finalmente, la cláusula **ORDER BY** ordena los resultados en función de la cantidad de aspirantes de forma descendente.