

Capacitación SQL

Nivel inicial

Unidad 2/8

Indice general

1. Estructura de una consulta SQL
2. Sintaxis SELECT
 - SQL SELECT
 - Comandos SQL más utilizados
 - Instrucción SQL SELECT DISTINCT
 - La instrucción SQL SELECT WHERE
3. Operadores aritméticos, lógicos y de comparación.
 - SQL AND, OR y NOT (lógicos)
 - Sintaxis de AND
 - Sintaxis de OR
 - Sintaxis de NOT
 - El operador SQL IN
 - Sintaxis de IN
 - El Operador SQL BETWEEN
 - El operador SQL LIKE
 - Alias SQL
 - Sintaxis de columna de alias
 - Sintaxis de tabla de alias
 - Operadores de Comparación

1. Estructura de una consulta SQL

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje diseñado para administrar información de los sistemas de gestión de bases de datos relacionales. Consiste en un lenguaje de definición de datos (DDL), un lenguaje de manipulación de datos (DML) y un lenguaje de control de datos (DCM). El alcance de SQL incluye la inserción de datos, consultas, actualizaciones, borrado, la creación y modificación de esquemas, como también el control de acceso a los datos.

DDL (Lenguaje de definición de datos):

Permite crear y modificar la estructura de una base de datos. Las sentencias que se utilizan son las siguientes:

CREATE: Utilizado para crear nuevas tablas, campos e índices.

ALTER: Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

DROP: Empleado para eliminar tablas e índices.

TRUNCATE: Empleado para eliminar todos los registros de una tabla.

COMMENT: Utilizado para agregar comentarios al diccionario de datos.

RENAME: Es utilizado para renombrar objetos.

DML (Lenguaje de Manipulación de datos):

Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una base de datos. Las sentencias que se utilizan son las siguientes:

SELECT: Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.

INSERT: Utilizado para cargar datos en la base de datos en una única operación.

UPDATE: Utilizado para modificar los valores de los campos y registros especificados

DELETE: Utilizado para eliminar registros de una tabla de una base de datos.

DCL (Lenguaje de control de datos):

Permite crear roles, permisos e integridad referencial, así como el control al acceso a la base de datos. Las sentencias que se utilizan son las siguientes:

GRANT: Utilizado para otorgar privilegios de acceso de usuario a la base de datos.

REVOKE: Utilizado para retirar privilegios de acceso otorgados con el comando GRANT.

2. Sintaxis SELECT

SQL SELECT

La instrucción SELECT se utiliza para seleccionar datos de una base de datos. Los datos devueltos se almacenan en una tabla de resultados, llamada conjunto de resultados.

Sintaxis:

```
SELECT column1, column2, ...
```

```
FROM table_name
```

Aquí, column1, column2, ...corresponden a los nombres de los campos de la tabla de la que desea seleccionar datos. Para el siguiente ejemplo, para obtener el listado de Clientes, se utiliza la tabla "**DimCustomer**" y se seleccionan los campos "**FirtsName**" y "**LastName**"

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

select
    FirstName,
    LastName
from DimCustomer
  
```

100 %

Results Messages

	FirstName	LastName
1	Jon	Yang
2	Eugene	Huang
3	Ruben	Torres
4	Christy	Zhu
5	Elizabeth	Johnson
6	Julio	Ruiz
7	Janet	Alvarez
8	Marco	Mehta
9	Rob	Verhoff
10	Shannon	Carlson
11	Jacquelyn	Suarez
12	Curtis	Lu
13	Lauren	Walker
14	Ian	Jenkins
15	Sydney	Bennett
16	Chloe	Young
17	Wyatt	Hill
18	Shannon	Wang
19	Clarence	Rai
20	Luke	Lal

✓ Query executed successfully.

A diferencia de la sentencia Select campo por campo, si se desea seleccionar todos los campos disponibles en la tabla, se utiliza la siguiente sintaxis:

```
SELECT * FROM table_name
```

En el siguiente ejemplo se utiliza la tabla **"DimCustomer"** con el objetivo de listar todos los campos almacenados para los Clientes.

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress	Year
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-works.com	90C
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adventure-works.com	60C
3	11002	31	AW00011002	NULL	Ruben	NULL	Torres	0	1971-02-09	M	NULL	M	ruben35@adventure-works.com	60C
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventure-works.com	70C
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@adventure-works.com	80C
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-works.com	70C
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-works.com	70C
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventure-works.com	60C
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-works.com	60C
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adventure-works.com	70C
11	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adventure-works.com	70C
12	11011	22	AW00011011	NULL	Curtis	NULL	Lu	0	1969-05-03	M	NULL	M	curtis9@adventure-works.com	60C
13	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@adventure-works.com	10C
14	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-works.com	10C
15	11014	634	AW00011014	NULL	Sydney	NULL	Bennett	0	1973-11-06	S	NULL	F	sydney23@adventure-works.com	10C
16	11015	301	AW00011015	NULL	Chloe	NULL	Young	0	1984-08-26	S	NULL	F	chloe23@adventure-works.com	30C
17	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventure-works.com	30C
18	11017	39	AW00011017	NULL	Shannon	NULL	Wang	0	1949-12-24	S	NULL	F	shannon1@adventure-works.com	20C
19	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adventure-works.com	30C
20	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventure-works.com	40C

A continuación se utiliza la misma tabla, **"DimCustomer"** pero con la sentencia **"select Top 10 * From DimCustomer"** El objetivo de esta consulta es que se liste únicamente los primeros 10 registros de Clientes.

Nota: la sentencia **"Top N"** arroja N resultados, generalmente los mejores o últimos dependiendo la sentencia, es decir un resultado reducido del contenido de la tabla o cualquier consulta. Esta consulta es muy útil para tener un set de datos de la tabla, ya que reduce el tiempo de ejecución de consulta. Esta sentencia es válida para SQL Server. Para otros motores de base de datos como Teradata, se utiliza **"sample X"**. Se recomienda revisar la sentencia dependiendo del motor de base de datos que se va a utilizar.

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```
select top 10 * from DimCustomer
```

100 %

Results Messages

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-work
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adventure
3	11002	31	AW00011002	NULL	Ruben	NULL	Torres	0	1971-02-09	M	NULL	M	ruben35@adventure-w
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventure-w
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@adventure
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-work
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-wor
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventure-w
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-work
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adventur

Comandos SQL más utilizados

SELECT	Extrae datos de una base de datos
UPDATE	Actualiza los datos en una base de datos
DELETE	Elimina datos de una base de datos
INSERT INTO	Inserta datos nuevos en una base de datos
CREATE DATABASE	Crea una nueva base de datos
ALTER DATABASE	Modifica una base de datos
CREATE TABLE	Crea una nueva tabla
ALTER TABLE	Modifica una tabla
DROP TABLE	Elimina una tabla
CREATE INDEX	Crea un índice (clave de búsqueda)
DROP INDEX	Elimina un índice

Instrucción SQL SELECT DISTINCT

La instrucción SELECT DISTINCT se utiliza para devolver solo valores distintos (diferentes).

Dentro de una tabla, una columna a menudo contiene valores duplicados y a veces solo se necesita enumerar los valores únicos, es decir sin duplicados. Con esta sentencia se especifica que los registros con ciertos datos duplicados sean obviadas en el resultado.

Sintaxis:

```
SELECT DISTINCT column1, column2, ...  
  
FROM table_name
```

La siguiente instrucción SQL selecciona solo los valores de la columna "FrenchOccupation" en la tabla "**DimCustomer**". En este caso listará todos los valores, aunque se encuentren duplicados.

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```
SELECT
  FrenchOccupation
FROM DimCustomer
```

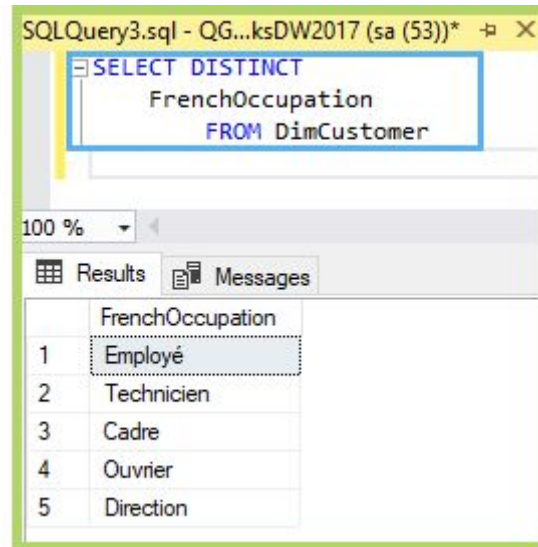
100 %

Results Messages

	FrenchOccupation
1	Cadre
2	Cadre
3	Cadre
4	Cadre
5	Cadre
6	Cadre
7	Cadre
8	Cadre
9	Cadre
10	Cadre
11	Cadre
12	Cadre
13	Direction
14	Direction
15	Direction
16	Technicien
17	Technicien
18	Technicien
19	Employé
20	Technicien
21	Technicien

✓ Query executed successfully.

La siguiente instrucción SQL selecciona solo los valores DISTINCT de la columna "FrenchOccupation" en la tabla **"DimCustomer"**: En este caso, se listarán todos los valores posibles que pueden ser seleccionados en el campo "FrenchOccupation"



La instrucción SQL SELECT WHERE

La instrucción SELECT WHERE se utiliza para extraer solo aquellos registros que cumplen una condición específica.

Sintaxis:

```
SELECT column1, column2, ...FROM table_name
```

```
WHERE condition
```

Si se necesita listar los clientes en los que el campo "**MiddleName**" no contenga nulos, se utiliza la siguiente sentencia:

Ejemplo con Nulos:

Nota: En el margen inferior derecho de la pantalla de resultado, se puede observar la cantidad de filas devueltas en base a la consulta (resaltada con naranja), en este caso, la consulta contiene 18484 filas.

SQLQuery3.sql - QG...ksDW2017 (sa (53))

```
SELECT * FROM DimCustomer
```

100 %

Results Messages

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@advent
3	11002	31	AW00011002	NULL	Ruben	NULL	Torres	0	1971-02-09	M	NULL	M	ruben35@adventu
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventu
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@advent
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-v
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventu
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-w
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adver
11	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adve
12	11011	22	AW00011011	NULL	Curtis	NULL	Lu	0	1969-05-03	M	NULL	M	curtis9@adventure
13	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@adventu
14	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-
15	11014	634	AW00011014	NULL	Sydney	NULL	Bennett	0	1973-11-06	S	NULL	F	sydney23@adventi
16	11015	301	AW00011015	NULL	Chloe	NULL	Young	0	1984-08-26	S	NULL	F	chloe23@adventur
17	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventur
18	11017	39	AW00011017	NULL	Shannon	NULL	Wang	0	1949-12-24	S	NULL	F	shannon1@advent
19	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adver
20	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventure

Query executed successfully. QG-SRV-CAP (14.0 RTM) sa (53) AdventureWorksDW2017 00:00:03 18484 rows

Ejemplos de funcionamiento de la instrucción where:

SQLQuery3.sql - QG...ksDW2017 (sa (53))

```
SELECT * FROM DimCustomer
WHERE MiddleName IS NOT NULL
```

100 %

Results Messages

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adver
3	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure
4	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adve
5	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adv
6	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@advent
7	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-
8	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventu
9	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adve
10	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventun
11	11020	53	AW00011020	NULL	Jordan	C	King	0	1984-03-19	S	NULL	M	jordan73@adventi
12	11022	609	AW00011022	NULL	Ethan	G	Zhang	0	1984-04-10	M	NULL	M	ethan20@adventu
13	11023	298	AW00011023	NULL	Seth	M	Edwards	0	1984-04-09	M	NULL	M	seth46@adventun
14	11027	40	AW00011027	NULL	Jessie	R	Zhao	0	1952-06-05	M	NULL	M	jessie16@adventu
15	11029	32	AW00011029	NULL	Jimmy	L	Moreno	0	1952-06-19	M	NULL	M	jimmy9@adventur
16	11030	28	AW00011030	NULL	Bethany	G	Yuan	0	1958-02-18	M	NULL	F	bethany10@advei
17	11031	8	AW00011031	NULL	Theresa	G	Ramos	0	1953-02-18	M	NULL	F	theresa13@adver
18	11036	343	AW00011036	NULL	Jennifer	C	Russell	0	1984-06-16	M	NULL	F	jennifer93@adven
19	11037	49	AW00011037	NULL	Chloe	M	Garcia	0	1983-05-27	S	NULL	F	chloe27@adventu
20	11038	6	AW00011038	NULL	Diana	F	Hernandez	0	1953-09-20	M	NULL	F	diana2@adventun

Query executed successfully. QG-SRV-CAP (14.0 RTM) sa (53) AdventureWorksDW2017 00:00:03 10654 rows

Con la instrucción SELECT WHERE observamos que la cantidad de filas del resultado de la consulta se redujo en número, ya que se excluyen los campos nulos.

Operadores aritméticos, lógicos y de comparación

Operadores SQL AND, OR y NOT (lógicos)

SÍMBOLO	SIGNIFICADO	EJEMPLO
No (Not)	No se cumple que	No A = B
y (And)	Y se cumple que	A = B y H > M
o (Or)	O se cumple que	A = B ó H > M

La cláusula WHERE puede combinarse con operadores AND, OR y NOT.

Los operadores AND y OR se utilizan con el objetivo de filtrar registros en función de más de una condición:

- El operador AND muestra un registro si todas las condiciones separadas por AND son VERDADERAS.
- El operador OR muestra un registro si alguna de las condiciones separadas por OR es VERDADERA.

El operador NOT muestra un registro si la (s) condición (es) NO ES VERDADERA.

Sintaxis del operador AND:

```
SELECT column1, column2, ...  
  
FROM table_name  
  
WHERE condition1 AND condition2 AND condition3 ...
```

La siguiente instrucción SQL selecciona todos los campos de la tabla **"DimCustomer"**

SQLQuery3.sql - QG...ksDW2017 (sa (53))

```
select * from DimCustomer
```

100 %

Results Messages

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress	Year
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-works.com	90C
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adventure-works.com	60C
3	11002	31	AW00011002	NULL	Ruben	NULL	Torres	0	1971-02-09	M	NULL	M	ruben35@adventure-works.com	60C
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventure-works.com	70C
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@adventure-works.com	80C
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-works.com	70C
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-works.com	70C
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventure-works.com	60C
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-works.com	60C
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adventure-works.com	70C
11	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adventure-works.com	70C
12	11011	22	AW00011011	NULL	Curtis	NULL	Lu	0	1969-05-03	M	NULL	M	curtis9@adventure-works.com	60C
13	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@adventure-works.com	10C
14	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-works.com	10C
15	11014	634	AW00011014	NULL	Sydney	NULL	Bennett	0	1973-11-06	S	NULL	F	sydney23@adventure-works.com	10C
16	11015	301	AW00011015	NULL	Chloe	NULL	Young	0	1984-08-26	S	NULL	F	chloe23@adventure-works.com	30C
17	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventure-works.com	30C
18	11017	39	AW00011017	NULL	Shannon	NULL	Wang	0	1949-12-24	S	NULL	F	shannon1@adventure-works.com	20C
19	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adventure-works.com	30C
20	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventure-works.com	40C

La siguiente instrucción SQL selecciona todos los registros de los campos **"FirtsName"** y **"LastName"** donde los valores del campo "SpanishEducation" se **"Licenciatura"** y los valores del campo "FrenchOccupation" sea **"Direction"** (en este caso sólo hay 1601 registros que cumplen dichas condiciones):

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE SpanishEducation = 'Licenciatura'
AND FrenchOccupation = 'Direction'
  
```

100 %

Results Messages

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Lauren	Walker	Licenciatura	Direction
2	Ian	Jenkins	Licenciatura	Direction
3	Sydney	Bennett	Licenciatura	Direction
4	Savannah	Baker	Licenciatura	Direction
5	Hunter	Davis	Licenciatura	Direction
6	Abigail	Price	Licenciatura	Direction
7	Beth	Jiménez	Licenciatura	Direction
8	Karla	Goel	Licenciatura	Direction
9	Ross	Vazquez	Licenciatura	Direction
10	Theodore	Gill	Licenciatura	Direction
11	Russell	Shen	Licenciatura	Direction
12	Jasmine	Barnes	Licenciatura	Direction
13	David	Rodriguez	Licenciatura	Direction
14	Bryce	Richardson	Licenciatura	Direction
15	Carol	Howard	Licenciatura	Direction
16	Jonathan	Hill	Licenciatura	Direction
17	Gabrielle	Adams	Licenciatura	Direction

✓ Query executed successfully.

Sintaxis del operador OR:

```

SELECT column1, column2, ...

FROM table_name

WHERE condition1 OR condition2 OR condition3 ...
  
```

La siguiente instrucción SQL selecciona todos los campos de la tabla **"DimCustomer"**, es decir, devuelve toda la información almacenada en la tabla de los clientes.



The screenshot shows a SQL query window with the text `select * from DimCustomer`. Below the query window, the 'Results' pane displays a table with 15 columns and 20 rows of data. The columns are: CustomerKey, GeographyKey, CustomerAlternateKey, Title, FirstName, MiddleName, LastName, NameStyle, BirthDate, MaritalStatus, Suffix, Gender, EmailAddress, and Year. The data represents a list of customers from the AdventureWorks database.

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress	Year
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-works.com	90C
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adventure-works.com	60C
3	11002	31	AW00011002	NULL	Ruben	NULL	Torres	0	1971-02-09	M	NULL	M	ruben35@adventure-works.com	60C
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventure-works.com	70C
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@adventure-works.com	80C
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-works.com	70C
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-works.com	70C
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventure-works.com	60C
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-works.com	60C
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adventure-works.com	70C
11	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adventure-works.com	70C
12	11011	22	AW00011011	NULL	Curtis	NULL	Lu	0	1969-05-03	M	NULL	M	curtis9@adventure-works.com	60C
13	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@adventure-works.com	10C
14	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-works.com	10C
15	11014	634	AW00011014	NULL	Sydney	NULL	Bennett	0	1973-11-06	S	NULL	F	sydney23@adventure-works.com	10C
16	11015	301	AW00011015	NULL	Chloe	NULL	Young	0	1984-08-26	S	NULL	F	chloe23@adventure-works.com	30C
17	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventure-works.com	30C
18	11017	39	AW00011017	NULL	Shannon	NULL	Wang	0	1949-12-24	S	NULL	F	shannon1@adventure-works.com	20C
19	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adventure-works.com	30C
20	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventure-works.com	40C

La siguiente instrucción SQL selecciona todos los registros del campo **"FirtsName"** y **"LastName"** donde el valor del campo FrenchOccupation sea **"Cadre"** el valor del campo "SpanishEducation" sea **"Licenciatura"** o **"Estudios universitarios (en curso)"** (en este caso hay 3548 registros que cumplan con dichas condiciones):

Nota: las sentencias que incluyen el operador **OR** tienen que encontrarse dentro de paréntesis, ya que indica que puede devolver resultados, tanto de un valor como el otro. De otra manera el resultado se verá alterado.

Ejemplo de consulta SQL sin la utilización de paréntesis:

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE FrenchOccupation = 'Cadre'
    and SpanishEducation = 'Licenciatura'
    or SpanishEducation = 'Estudios Universitarios (en curso)'
    
```

100 %

Results Messages

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Jon	Yang	Licenciatura	Cadre
2	Eugene	Huang	Licenciatura	Cadre
3	Ruben	Tomes	Licenciatura	Cadre
4	Christy	Zhu	Licenciatura	Cadre
5	Elizabeth	Johnson	Licenciatura	Cadre
6	Julio	Ruiz	Licenciatura	Cadre
7	Janet	Alvarez	Licenciatura	Cadre
8	Marco	Mehta	Licenciatura	Cadre
9	Rob	Verhoff	Licenciatura	Cadre
10	Shannon	Carlson	Licenciatura	Cadre
11	Jacquelyn	Suarez	Licenciatura	Cadre
12	Curtis	Lu	Licenciatura	Cadre
13	Chloe	Young	Estudios universitarios (en curso)	Technicien
14	Wyatt	Hill	Estudios universitarios (en curso)	Technicien
15	Clarence	Rai	Estudios universitarios (en curso)	Employé
16	Destiny	Wilson	Estudios universitarios (en curso)	Technicien
17	Ethan	Zhang	Estudios universitarios (en curso)	Technicien

Query executed successfully. QG-SRV-CAP

Ejemplo de consulta SQL con utilización de paréntesis:

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE FrenchOccupation = 'Cadre'
and (SpanishEducation = 'Licenciatura'
or SpanishEducation = 'Estudios Universitarios (en curso)')
    
```

100 %

Results Messages

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Jon	Yang	Licenciatura	Cadre
2	Eugene	Huang	Licenciatura	Cadre
3	Ruben	Torres	Licenciatura	Cadre
4	Christy	Zhu	Licenciatura	Cadre
5	Elizabeth	Johnson	Licenciatura	Cadre
6	Julio	Ruiz	Licenciatura	Cadre
7	Janet	Alvarez	Licenciatura	Cadre
8	Marco	Mehta	Licenciatura	Cadre
9	Rob	Verhoff	Licenciatura	Cadre
10	Shannon	Carlson	Licenciatura	Cadre
11	Jacquelyn	Suarez	Licenciatura	Cadre
12	Curtis	Lu	Licenciatura	Cadre
13	Grace	Butler	Estudios universitarios (en curso)	Cadre
14	Caleb	Carter	Estudios universitarios (en curso)	Cadre
15	Ryan	Brown	Estudios universitarios (en curso)	Cadre
16	Tamara	Liang	Estudios universitarios (en curso)	Cadre
17	Trevor	Bryant	Estudios universitarios (en curso)	Cadre

✓ Query executed successfully. QG-SRV-CAP

Sintaxis de NOT:

```
SELECT column1, column2, ...
```

```
FROM table_name
```

WHERE NOT condition

La siguiente instrucción SQL selecciona todos los campos de la tabla **"DimCustomer"**

SQLQuery3.sql - QG...ksDW2017 (sa (53))

```
select * from DimCustomer
```

100 %

Results Messages

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress	Year
1	11000	26	AW00011000	NULL	Jon	V	Yang	0	1971-10-06	M	NULL	M	jon24@adventure-works.com	90C
2	11001	37	AW00011001	NULL	Eugene	L	Huang	0	1976-05-10	S	NULL	M	eugene10@adventure-works.com	60C
3	11002	31	AW00011002	NULL	Ruben	NULL	Tomes	0	1971-02-09	M	NULL	M	ruben35@adventure-works.com	60C
4	11003	11	AW00011003	NULL	Christy	NULL	Zhu	0	1973-08-14	S	NULL	F	christy12@adventure-works.com	70C
5	11004	19	AW00011004	NULL	Elizabeth	NULL	Johnson	0	1979-08-05	S	NULL	F	elizabeth5@adventure-works.com	80C
6	11005	22	AW00011005	NULL	Julio	NULL	Ruiz	0	1976-08-01	S	NULL	M	julio1@adventure-works.com	70C
7	11006	8	AW00011006	NULL	Janet	G	Alvarez	0	1976-12-02	S	NULL	F	janet9@adventure-works.com	70C
8	11007	40	AW00011007	NULL	Marco	NULL	Mehta	0	1969-11-06	M	NULL	M	marco14@adventure-works.com	60C
9	11008	32	AW00011008	NULL	Rob	NULL	Verhoff	0	1975-07-04	S	NULL	F	rob4@adventure-works.com	60C
10	11009	25	AW00011009	NULL	Shannon	C	Carlson	0	1969-09-29	S	NULL	M	shannon38@adventure-works.com	70C
11	11010	22	AW00011010	NULL	Jacquelyn	C	Suarez	0	1969-08-05	S	NULL	F	jacquelyn20@adventure-works.com	70C
12	11011	22	AW00011011	NULL	Curtis	NULL	Lu	0	1969-05-03	M	NULL	M	curtis9@adventure-works.com	60C
13	11012	611	AW00011012	NULL	Lauren	M	Walker	0	1979-01-14	M	NULL	F	lauren41@adventure-works.com	10C
14	11013	543	AW00011013	NULL	Ian	M	Jenkins	0	1979-08-03	M	NULL	M	ian47@adventure-works.com	10C
15	11014	634	AW00011014	NULL	Sydney	NULL	Bennett	0	1973-11-06	S	NULL	F	sydney23@adventure-works.com	10C
16	11015	301	AW00011015	NULL	Chloe	NULL	Young	0	1984-08-26	S	NULL	F	chloe23@adventure-works.com	30C
17	11016	329	AW00011016	NULL	Wyatt	L	Hill	0	1984-10-25	M	NULL	M	wyatt32@adventure-works.com	30C
18	11017	39	AW00011017	NULL	Shannon	NULL	Wang	0	1949-12-24	S	NULL	F	shannon1@adventure-works.com	20C
19	11018	32	AW00011018	NULL	Clarence	D	Rai	0	1955-10-06	S	NULL	M	clarence32@adventure-works.com	30C
20	11019	52	AW00011019	NULL	Luke	L	Lal	0	1983-09-04	S	NULL	M	luke18@adventure-works.com	40C

La siguiente instrucción SQL selecciona todos los valores de los campos **"FirtsName"** y **"LastName"** donde el valor del campo FrenchOccupation NO sea **"Cadre"** y al mismo tiempo el valor del campo SpanishEducation sea **"Licenciatura" o "Estudios universitarios (en curso)"** (en este caso existen 8532 registros que cumplan con dichas condiciones):

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE NOT FrenchOccupation = 'Cadre'
and SpanishEducation = 'Licenciatura'
or SpanishEducation = 'Estudios Universitarios (en curso)'
    
```

100 %

Results Messages

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Lauren	Walker	Licenciatura	Direction
2	Ian	Jenkins	Licenciatura	Direction
3	Sydney	Bennett	Licenciatura	Direction
4	Chloe	Young	Estudios universitarios (en curso)	Technicien
5	Wyatt	Hill	Estudios universitarios (en curso)	Technicien
6	Clarence	Rai	Estudios universitarios (en curso)	Employé
7	Destiny	Wilson	Estudios universitarios (en curso)	Technicien
8	Ethan	Zhang	Estudios universitarios (en curso)	Technicien
9	Seth	Edwards	Estudios universitarios (en curso)	Technicien
10	Russell	Xie	Estudios universitarios (en curso)	Technicien
11	Harold	Sai	Estudios universitarios (en curso)	Employé
12	Jessie	Zhao	Estudios universitarios (en curso)	Employé
13	Jill	Jimenez	Estudios universitarios (en curso)	Employé
14	Jimmy	Moreno	Estudios universitarios (en curso)	Employé
15	Jennifer	Russell	Estudios universitarios (en curso)	Technicien
16	Marc	Martin	Estudios universitarios (en curso)	Employé
17	Jesse	Murphy	Estudios universitarios (en curso)	Technicien

Query executed successfully. QG-SRV-CAP

Sintaxis del operador IN

El operador IN permite especificar múltiples valores en una cláusula WHERE. Este operador es una abreviatura para múltiples condiciones OR.

Sintaxis:

```

SELECT column_name(s)

FROM table_name

WHERE column_name IN (value1, value2, ...)
    
```

La siguiente instrucción SQL selecciona todos los registros de los campos **"FirtsName"** y **"LastName"** donde el valor del campo "FrenchOccupation" sea **"Cadre"** o **"Direction"** y al mismo tiempo el valor del campo "SpanishEducation" sea **"Licenciatura"** (en este caso existen 3489 registros que cumplan con dichas condiciones):

The screenshot shows a SQL query window titled "SQLQuery3.sql - QG...ksDW2017 (sa (53))*". The query is as follows:

```
SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE FrenchOccupation IN ('Cadre','Direction')
AND SpanishEducation = 'Licenciatura'
```

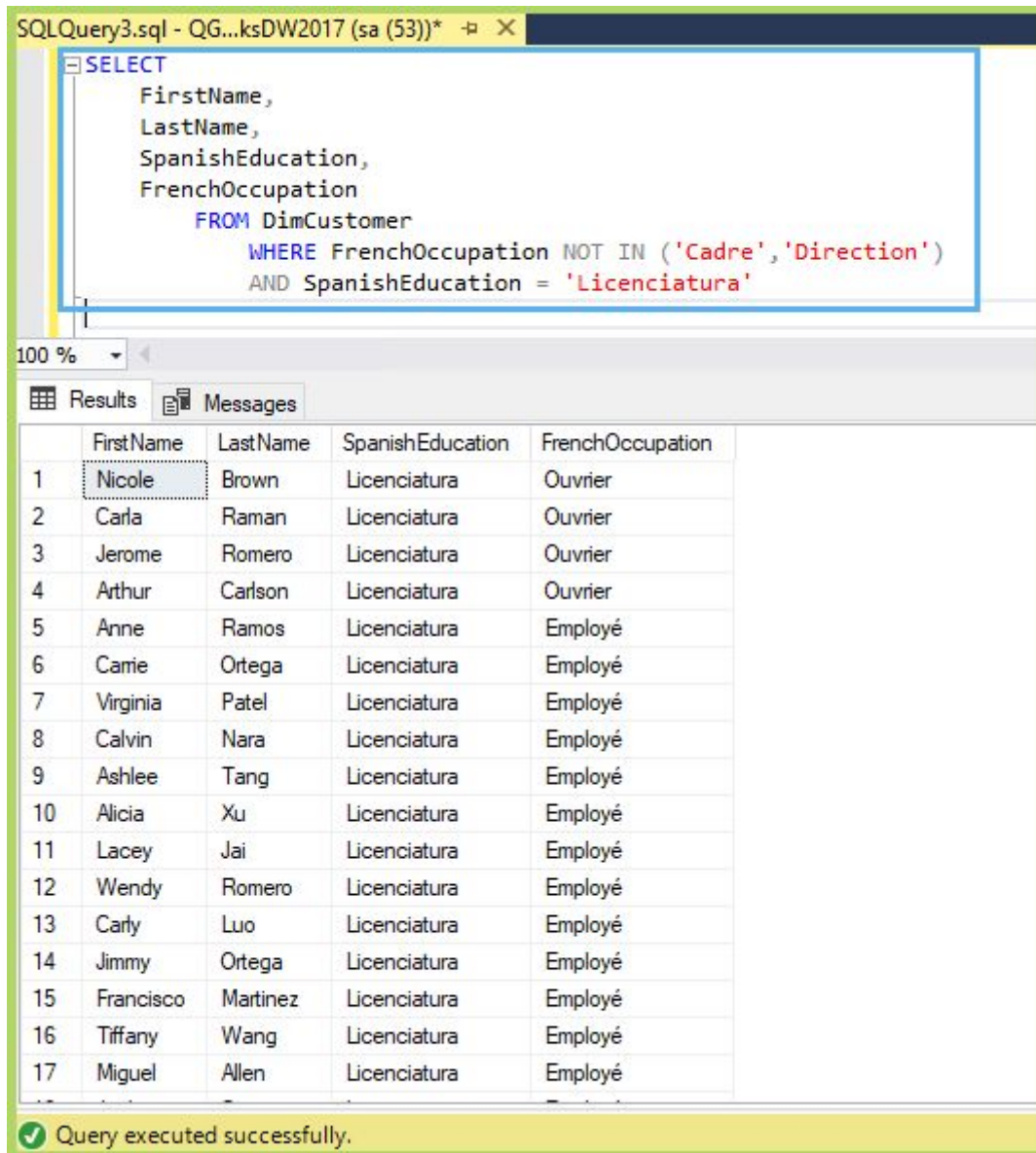
Below the query window, the "Results" tab is active, displaying a table with 5 columns: "FirstName", "LastName", "SpanishEducation", and "FrenchOccupation". The table contains 17 rows of data. The first row is highlighted.

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Jon	Yang	Licenciatura	Cadre
2	Eugene	Huang	Licenciatura	Cadre
3	Ruben	Torres	Licenciatura	Cadre
4	Christy	Zhu	Licenciatura	Cadre
5	Elizabeth	Johnson	Licenciatura	Cadre
6	Julio	Ruiz	Licenciatura	Cadre
7	Janet	Alvarez	Licenciatura	Cadre
8	Marco	Mehta	Licenciatura	Cadre
9	Rob	Verhoff	Licenciatura	Cadre
10	Shannon	Carlson	Licenciatura	Cadre
11	Jacquelyn	Suarez	Licenciatura	Cadre
12	Curtis	Lu	Licenciatura	Cadre
13	Lauren	Walker	Licenciatura	Direction
14	Ian	Jenkins	Licenciatura	Direction
15	Sydney	Bennett	Licenciatura	Direction
16	Savannah	Baker	Licenciatura	Direction
17	Hunter	Davis	Licenciatura	Direction

At the bottom of the window, a status bar indicates: "Query executed successfully."

La siguiente instrucción SQL selecciona todos los registros de los campos **"FirtsName"** y **"LastName"** donde el valor del campo "FrenchOccupation" NO sea **"Cadre"** o **"Direction"** y al mismo tiempo el valor del campo "SpanishEducation"

sea **"Licenciatura"** (en este caso existen 1867 registros que cumplan con dichas condiciones):



The screenshot shows a SQL query window titled "SQLQuery3.sql - QG...ksDW2017 (sa (53))". The query is as follows:

```
SELECT
    FirstName,
    LastName,
    SpanishEducation,
    FrenchOccupation
FROM DimCustomer
WHERE FrenchOccupation NOT IN ('Cadre','Direction')
AND SpanishEducation = 'Licenciatura'
```

Below the query editor, the "Results" tab is active, displaying a table with 17 rows and 5 columns: FirstName, LastName, SpanishEducation, and FrenchOccupation. The first row is highlighted.

	FirstName	LastName	SpanishEducation	FrenchOccupation
1	Nicole	Brown	Licenciatura	Ouvrier
2	Carla	Raman	Licenciatura	Ouvrier
3	Jerome	Romero	Licenciatura	Ouvrier
4	Arthur	Carlson	Licenciatura	Ouvrier
5	Anne	Ramos	Licenciatura	Employé
6	Camie	Ortega	Licenciatura	Employé
7	Virginia	Patel	Licenciatura	Employé
8	Calvin	Nara	Licenciatura	Employé
9	Ashlee	Tang	Licenciatura	Employé
10	Alicia	Xu	Licenciatura	Employé
11	Lacey	Jai	Licenciatura	Employé
12	Wendy	Romero	Licenciatura	Employé
13	Carly	Luo	Licenciatura	Employé
14	Jimmy	Ortega	Licenciatura	Employé
15	Francisco	Martinez	Licenciatura	Employé
16	Tiffany	Wang	Licenciatura	Employé
17	Miguel	Allen	Licenciatura	Employé

At the bottom of the window, a green status bar indicates: "Query executed successfully."

Sintaxis del operador BETWEEN

El operador BETWEEN selecciona valores dentro de un rango determinado. Los valores incluidos pueden ser números, texto o fechas.

Nota: El operador BETWEEN es inclusivo, es decir, se incluyen valores iniciales y finales. Si se indica en la sentencia "WHERE column_name BETWEEN 10 AND 20" los valores entre '10' y '20' se incluyen en el resultado.

Sintaxis:

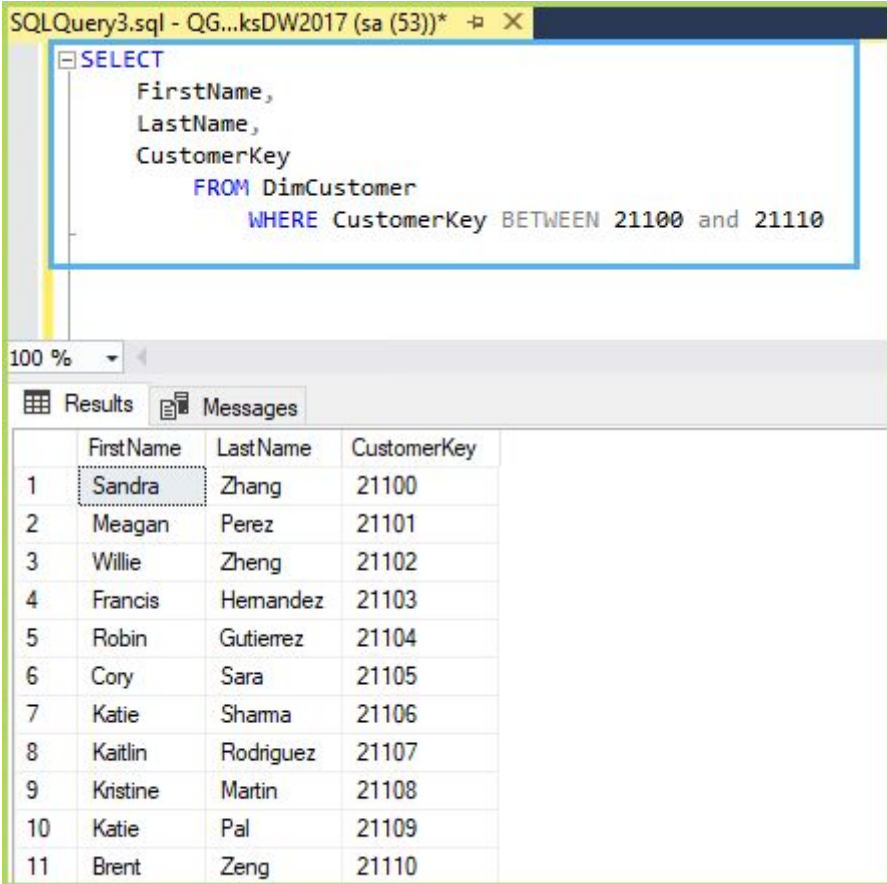
```

SELECT column_name(s)

FROM table_name

WHERE column_name BETWEEN value1 AND value2
  
```

La siguiente instrucción SQL selecciona todos los valores de los campos **"FirtsName"** y **"LastName"** donde el valor del campo "CustomerKey" se encuentra entre 21100 y 21110 (en este caso existen 11 registros que cumplan con dichas condiciones):



The screenshot shows a SQL query window titled 'SQLQuery3.sql - QG...ksDW2017 (sa (53))'. The query is as follows:

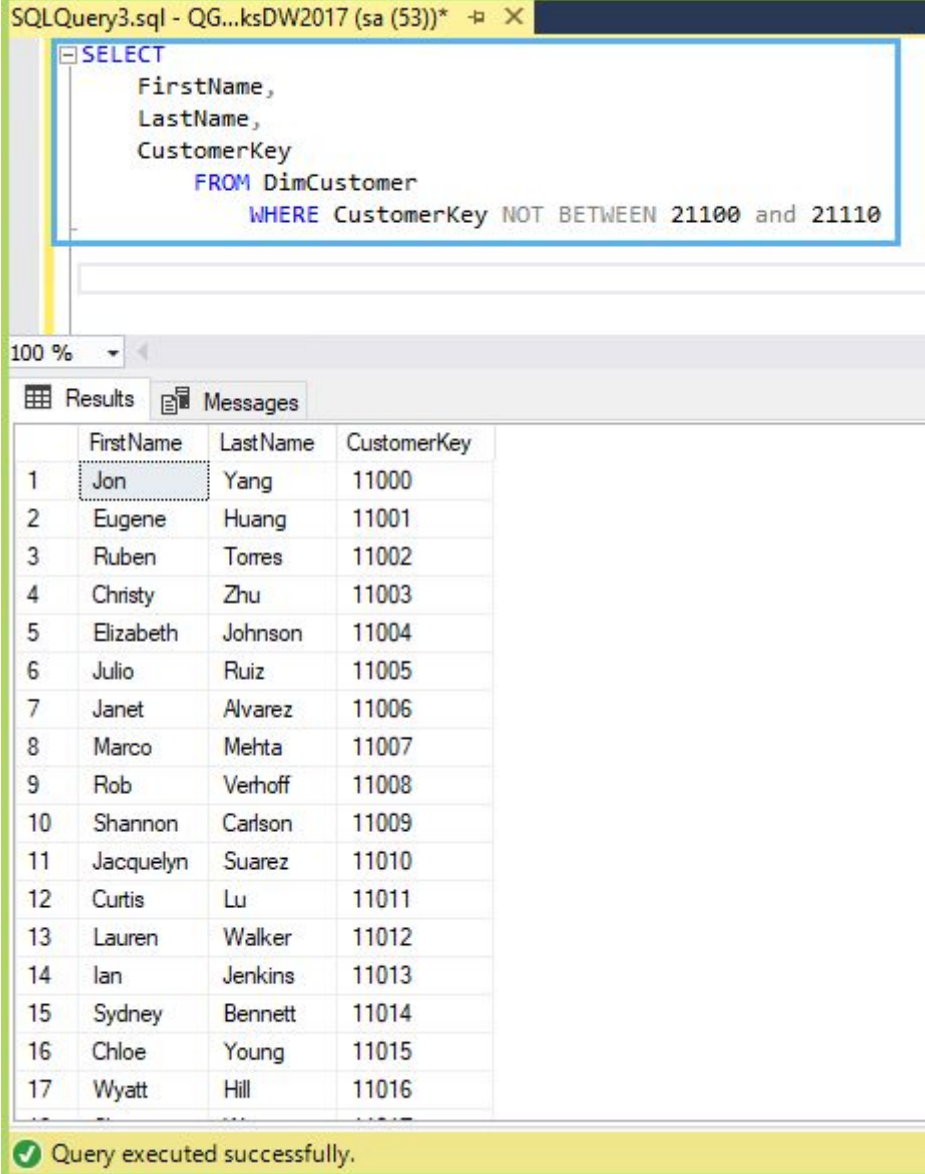
```

SELECT
    FirstName,
    LastName,
    CustomerKey
FROM DimCustomer
WHERE CustomerKey BETWEEN 21100 and 21110
  
```

Below the query window, the 'Results' tab is active, displaying 11 rows of data. The first row is highlighted. The columns are 'FirstName', 'LastName', and 'CustomerKey'.

	FirstName	LastName	CustomerKey
1	Sandra	Zhang	21100
2	Meagan	Perez	21101
3	Willie	Zheng	21102
4	Francis	Hernandez	21103
5	Robin	Gutierrez	21104
6	Cory	Sara	21105
7	Katie	Sharma	21106
8	Kaitlin	Rodriguez	21107
9	Kristine	Martin	21108
10	Katie	Pal	21109
11	Brent	Zeng	21110

Si se desea visualizar los Customers que se encuentran fuera del rango del ejemplo anterior, es necesario utilizar NO BETWEEN (en este caso a diferencia del ejemplo anterior, existen 18573 registros que cumplan con dichas condiciones)



The screenshot shows a SQL query window titled "SQLQuery3.sql - QG...ksDW2017 (sa (53))*". The query is as follows:

```
SELECT
    FirstName,
    LastName,
    CustomerKey
FROM DimCustomer
WHERE CustomerKey NOT BETWEEN 21100 and 21110
```

Below the query window, the "Results" tab is active, displaying a table with 17 rows and 4 columns: FirstName, LastName, and CustomerKey. The first row is highlighted.

	FirstName	LastName	CustomerKey
1	Jon	Yang	11000
2	Eugene	Huang	11001
3	Ruben	Torres	11002
4	Christy	Zhu	11003
5	Elizabeth	Johnson	11004
6	Julio	Ruiz	11005
7	Janet	Alvarez	11006
8	Marco	Mehta	11007
9	Rob	Verhoff	11008
10	Shannon	Carlson	11009
11	Jacquelyn	Suarez	11010
12	Curtis	Lu	11011
13	Lauren	Walker	11012
14	Ian	Jenkins	11013
15	Sydney	Bennett	11014
16	Chloe	Young	11015
17	Wyatt	Hill	11016

At the bottom of the window, a status bar indicates: "Query executed successfully."

Operador BETWEEN utilizando IN:

La siguiente instrucción SQL selecciona todos registros de los campos "FirtsName" y "LastName" donde el valor del campo "CustomerKey" se encuentre entre 21100 y 21110 y además, el valor del campo "FrenchOccupation" no sea "Cadre", "Direction" ni "Technicien" (en este caso existen 5308 registros que cumplan con dichas condiciones):

SQLQuery3.sql - QG...ksDW2017 (sa (53))*

```

SELECT
    FirstName,
    LastName,
    CustomerKey,
    FrenchOccupation
FROM DimCustomer
WHERE CustomerKey NOT BETWEEN 21100 and 21110
AND FrenchOccupation NOT IN ('Cadre', 'Direction', 'Technicien')
Order By FrenchOccupation
    
```

100 %

Results Messages

	FirstName	LastName	CustomerKey	FrenchOccupation
1	Clarence	Rai	11018	Employé
2	Alejandro	Beck	11025	Employé
3	Harold	Sai	11026	Employé
4	Jessie	Zhao	11027	Employé
5	Jill	Jimenez	11028	Employé
6	Jimmy	Moreno	11029	Employé
7	Bethany	Yuan	11030	Employé
8	Wendy	Dominguez	11035	Employé
9	Chloe	Garcia	11037	Employé
10	Diana	Hernandez	11038	Employé
11	Marc	Martin	11039	Employé
12	Adam	Flores	11044	Employé
13	Carol	Rai	11049	Employé
14	Heidi	Lopez	11052	Employé
15	Evan	James	11119	Employé
16	Angela	Griffin	11133	Employé
17	Marcus	Harris	11135	Employé

Query executed successfully. QG-SRV-CAP

Nota: la sentencia “Order by” arroja resultados ordenados por el campo indicado, generalmente los resultados sin el operador “Order by” no siguen un criterio en el orden de visualizar los registros. Es posible indicar en la misma sentencia que el orden se realice de forma ascendente o descendente (asc, desc). En este caso ordena Apellido por orden alfabético.

La siguiente instrucción SQL selecciona todos los registros de los campos **"FirtsName"** y **"LastName"** donde el valor del campo "BirthDate" se encuentre ordenado de forma descendente:

SQLQuery1.sql - qg..._dev_user_dw (98))*

```

Select
    FirstName,
    LastName,
    BirthDate
FROM DimCustomer
WHERE BirthDate BETWEEN '1980-01-01' AND '1980-03-4'
Order By BirthDate desc
    
```

100 %

Results Messages

	FirstName	LastName	BirthDate
1	Kirk	Nason	1980-03-04
2	Devin	Coleman	1980-03-03
3	Christian	Henderson	1980-03-03
4	Jillian	Lopez	1980-03-03
5	Anne	Blanco	1980-03-03
6	Isabella	Kelly	1980-03-03
7	Evan	Turner	1980-03-03
8	Caitlin	Howard	1980-03-03
9	Lucas	Hughes	1980-03-03
10	Martha	Zeng	1980-02-23
11	Megan	Wood	1980-02-22
12	Clayton	Yang	1980-02-22
13	Connor	Mitchell	1980-02-21
14	Audrey	Alvarez	1980-02-21
15	Darren	Ruiz	1980-02-21
16	Noah	Phillips	1980-02-20
17	Kristine	Serrano	1980-02-20
18	Marie	Munoz	1980-02-20
19	Kenneth	Pal	1980-02-20
20	Mallory	Romero	1980-02-20
21	Cathy	Adams	1980-02-20

Query executed successfully. | qg-db-dev.

Sintaxis del operador LIKE

El operador LIKE se utiliza en una cláusula WHERE con el objetivo de buscar un patrón específico en una columna.

Existen dos caracteres comodines que se utilizan con frecuencia junto con el operador LIKE:

- %: El signo de porcentaje representa cero, uno o varios caracteres
- _ : El guión bajo representa un solo carácter

Sintaxis:

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE columnN LIKE pattern
```

Nota: Se recomienda usar la sentencia LIKE con una función de LOWER() o UPPER(), estas mismas se estudiarán en profundidad en clases siguientes. Estas funciones se utilizan para transformar las cadenas de texto en minúsculas o mayúsculas, respectivamente, para evitar que el resultado de la consulta no tenga en cuenta valores con variables de mayúsculas o minúsculas.

La siguiente instrucción SQL selecciona a todos los Customer con el valor del campo "LastName" que comienza con "a" . Se utiliza la función LOWER() con el objetivo de realizar la búsqueda en la cadena de un carácter en minúscula por lo que es necesario que la cadena se encuentre en minúscula para que la ocurrencia de la condición sea precisa, ordenado por el campo "FirstName".

SQLQuery1.sql - qq..._dev_user_dw (98))*

```

Select
    FirstName,
    LastName
FROM DimCustomer
WHERE LastName LIKE LOWER ('a%')
Order By FirstName
    
```

100 %

Results Messages

	FirstName	LastName
1	Aaron	Allen
2	Aaron	Alexander
3	Aaron	Adams
4	Abby	Arthur
5	Abigail	Anderson
6	Abigail	Alexander
7	Adam	Allen
8	Adam	Adams
9	Adriana	Arthur
10	Adrienne	Avarez
11	Aidan	Alexander
12	Albert	Alonso
13	Albert	Avarez
14	Alberto	Avarez
15	Alberto	Alonso
16	Alejandro	Anand
17	Alejandro	Andersen
18	Alex	Adams
19	Alex	Allen
20	Alexander	Anderson
21	Alexander	Allen

✓ Query executed successfully.

La siguiente instrucción SQL selecciona a todos los Customer cuyos valores en el campo "LastName" finalizan con "a":

SQLQuery1.sql - qq..._dev_user_dw (98))*

```

Select
    FirstName,
    LastName
FROM DimCustomer
WHERE LastName LIKE LOWER ('%a')
Order By FirstName
  
```

100 %

Results Messages

	FirstName	LastName
1	Aaron	Sharma
2	Abby	Rana
3	Abby	Chandra
4	Abby	Mehta
5	Abby	Garcia
6	Abigail	Garcia
7	Abigail	Rivera
8	Adam	Sharma
9	Adrian	Rivera
10	Adriana	Rana
11	Adriana	Garcia
12	Adriana	Sara
13	Adriana	Chandra
14	Adriana	Mehta
15	Adriana	Malhotra
16	Adrienne	Ortega
17	Aimee	Ma
18	Alan	Ma
19	Albert	Ortega
20	Alberto	Ortega
21	Alfred	Ma

Query executed successfully.

La siguiente instrucción SQL selecciona a todos los Customer en los que los valores del campo “LastName” contengan "sh":

SQLQuery1.sql - qg..._dev_user_dw (98))*

```

Select
    FirstName,
    LastName
FROM DimCustomer
WHERE LastName LIKE LOWER ('%sh%')
Order By FirstName
  
```

100 %

Results Messages

	FirstName	LastName
1	Aaron	Shama
2	Aaron	Shan
3	Aaron	Washington
4	Abigail	Washington
5	Adam	Washington
6	Adam	Shama
7	Adam	Shan
8	Aidan	Washington
9	Aimee	She
10	Alan	She
11	Alan	Shen
12	Alejandro	Shan
13	Alejandro	Shama
14	Alejandro	Shen
15	Alejandro	She
16	Alexandra	Washington
17	Alexandria	Washington
18	Alexia	Washington
19	Alexis	Ashe
20	Alexis	Washington
21	Alexis	Shama

✓ Query executed successfully.

La siguiente instrucción SQL selecciona a todos los Customer en los que los valores del campo “LastName” contengan en su segunda letra el carácter "e":

The screenshot shows a SQL Query Editor window titled "SQLQuery1.sql - qg..._dev_user_dw (98)". The query is as follows:

```

Select
    FirstName,
    LastName
FROM DimCustomer
WHERE LastName LIKE LOWER ('_e')
Order By FirstName
  
```

Below the query editor, the "Results" tab is active, displaying a table with 20 rows. The first row is highlighted. The table columns are "FirstName" and "LastName".

	FirstName	LastName
1	Aimee	He
2	Alan	He
3	Alan	Ye
4	Alejandro	Ye
5	Alisha	He
6	Alisha	Ye
7	Alvin	He
8	Alvin	Ye
9	Amy	Ye
10	Amy	He
11	Arturo	He
12	Arturo	Ye
13	Autumn	He
14	Barbara	Ye
15	Barbara	He
16	Brent	Ye
17	Brent	He
18	Brittney	Ye
19	Brittney	He
20	Candice	He

At the bottom of the window, a status bar indicates: "Query executed successfully."

Nota: El signo de porcentaje (%) y el guión bajo (_) también se pueden utilizar en combinaciones. Se puede combinar cualquier cantidad de condiciones utilizando los operadores AND y OR

Algunos ejemplos que muestran diferentes operadores LIKE con comodines '%' y '_':

Like Operador	Descripción
<code>WHERE column1 LIKE 'a%'</code>	Encuentra cualquier valor que empiece con 'a'
<code>WHERE column1 LIKE '%a'</code>	Encuentra cualquier valor que termine con 'a'
<code>WHERE column1 LIKE '%ar%'</code>	Encuentra cualquier valor que tenga 'ar' en algún lado
<code>WHERE column1 LIKE '_r%'</code>	Encuentra cualquier valor que tenga 'r' en segundo lugar
<code>WHERE column1 LIKE 'a__%'</code>	Encuentra cualquier valor que empiece con 'a' y tenga 3 caracteres de largo
<code>WHERE column1 LIKE 'a%o'</code>	Encuentra cualquier valor que empiece con 'a' y termine con 'o'

Alias SQL

Los alias de SQL se utilizan para asignar a una tabla, o una columna en una tabla, un nombre temporal.

Los alias a menudo se utilizan para que los nombres de columna sean más legibles.

Solo existe un alias mientras dura la consulta.

Sintaxis de columna de alias:

```
SELECT column_name AS alias_name
FROM table_name
```

Sintaxis de tabla de alias:

```
SELECT column_name(s)
FROM table_name AS alias_name
```


Nota: Requiere comillas dobles o corchetes si el nombre del alias contiene espacios.

SQLQuery1.sql - qq..._dev_user_dw (98))

```

Select
    FirstName AS Nombre,
    LastName AS Apellido,
    BirthDate AS "Fecha de Nacimiento"
FROM DimCustomer
Order By BirthDate
    
```

100 %

Results Messages

	Nombre	Apellido	Fecha de Nacimiento
1	Gabrielle	James	1916-02-10
2	Deb	Moreno	1917-02-09
3	Franklin	Pal	1917-06-05
4	Ethan	Robinson	1917-09-20
5	Dale	Andersen	1918-02-11
6	Robert	Johnson	1918-11-08
7	Vanessa	Hayes	1919-02-14
8	Emma	Williams	1919-03-10
9	Kelly	Wood	1920-11-14
10	Jeremy	Brooks	1922-01-02
11	Brianna	Sandberg	1922-04-10
12	Haley	Allen	1922-06-06
13	Mariah	Bennett	1923-04-14
14	Wyatt	Parker	1923-08-16
15	Angel	Murphy	1923-11-16
16	Jaclyn	Wu	1924-10-05
17	Jack	Coleman	1924-12-03
18	Mackenzie	Baker	1925-02-23
19	Elizabeth	Perry	1925-08-25
20	Sharon	Goel	1926-01-10
21	Alma	Kelly	1926-02-15

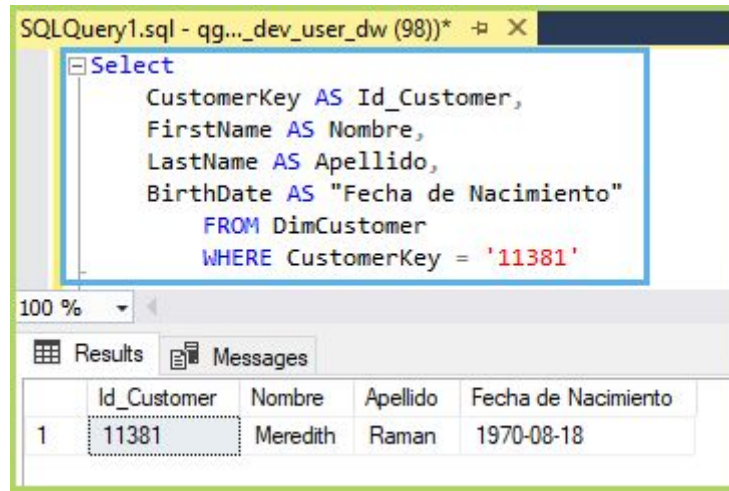
✓ Query executed successfully.

Operadores de Comparación

Suponiendo que la variable a tiene un valor de 5 y la variable b tiene un valor de 10

Operador	Descripción	Ejemplo
=	Compara dos valores	(a = b) es falso.
!=	Compara si son diferentes	(a != b) es verdadero.
<>	Diferente a	(a <> b) es verdadero
>	Mayor que	(a > b) es falso.
<	Menor que	(a < b) es verdadero.
>=	Mayor o igual que	(a >= b) es falso.
<=	Menor o igual que	(a <= b) es verdadero.
!<	No es menor que	(a !< b) es falso.
!>	No es mayor que	(a !> b) es verdadero.

La siguiente instrucción de SQL, muestra un ejemplo con un operador de comparación entre el campo **"CustomerKey"** y el número específico de Id que se desea obtener:



The screenshot shows a SQL query window titled 'SQLQuery1.sql - qq..._dev_user_dw (98))' with a query that selects customer information from the 'DimCustomer' table. Below the query editor, the 'Results' tab is active, displaying a single row of data for a customer with ID 11381.

```

Select
    CustomerKey AS Id_Customer,
    FirstName AS Nombre,
    LastName AS Apellido,
    BirthDate AS "Fecha de Nacimiento"
FROM DimCustomer
WHERE CustomerKey = '11381'
  
```

	Id_Customer	Nombre	Apellido	Fecha de Nacimiento
1	11381	Meredith	Raman	1970-08-18

La siguiente instrucción SQL muestra un ejemplo con un operador de comparación entre un campo de la tabla **"DimCustomer" (BirthDate)** y una función que arroja la fecha de hoy (más adelante se tratarán esas funciones a detalle) Arrojando un resultado donde indica que el valor del campo **"BirthDate"** tiene que ser anterior al día de hoy.

SQLQuery1.sql - qg..._dev_user_dw (98))*

```

Select
    CustomerKey AS Id_Customer,
    FirstName AS Nombre,
    LastName AS Apellido,
    BirthDate AS "Fecha de Nacimiento"
FROM DimCustomer
WHERE BirthDate < GETDATE()
  
```

100 %

Results Messages

	Id_Customer	Nombre	Apellido	Fecha de Nacimiento
1	11000	Jon	Yang	1971-10-06
2	11001	Eugene	Huang	1976-05-10
3	11002	Ruben	Torres	1971-02-09
4	11003	Christy	Zhu	1973-08-14
5	11004	Elizabeth	Johnson	1979-08-05
6	11005	Julio	Ruiz	1976-08-01
7	11006	Janet	Alvarez	1976-12-02
8	11007	Marco	Mehta	1969-11-06
9	11008	Rob	Verhoff	1975-07-04
10	11009	Shannon	Carlson	1969-09-29
11	11010	Jacquelyn	Suarez	1969-08-05
12	11011	Curtis	Lu	1969-05-03
13	11012	Lauren	Walker	1979-01-14
14	11013	Ian	Jenkins	1979-08-03
15	11014	Sydney	Bennett	1973-11-06
16	11015	Chloe	Young	1984-08-26
17	11016	Wyatt	Hill	1984-10-25
18	11017	Shannon	Wang	1949-12-24
19	11018	Clarence	Rai	1955-10-06

✓ Query executed successfully.

Nota: La función Getdate() devuelve como resultado la fecha y hora actuales

----- *Seguí reforzando tu conocimiento en la próxima unidad del programa.