

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS, 2024-1

FUNDAMENTOS DE BASES DE DATOS



Proyecto Final:

Zoológico de Huitziltepec.

PROFESOR:

Gerardo Avilés Rosas

INTEGRANTES DEL EQUIPO:

Aguirre Morales Gael Alejandro
Fernández Blancas Melissa Lizbeth
López Carrillo Alan Ignacio
Sánchez Nava Rodrigo
Sánchez Salmerón Ethan Damián

CONSULTAS SQL PARA LA BASE DE DATOS “EL ZOOLOGICO DE HUITZILTEPEC”

1. Encontrar los proveedores que han suministrado medicina a biomas donde hay más de 3 animales. Mostrar la siguiente información en la consulta: nombre y apellidos del proveedor, la medicina suministrada, el tipo del bioma al que se le suministra, y la cantidad de animales que hay en el bioma.

```
SELECT P.Nombre, P.ApellidoPaterno, M.Nombre AS MedicinaSuministrada, B.TipoBioma,
COUNT(A.IDAnimal) AS CantidadAnimales
FROM Proveedor P
JOIN ProveerMedicina PM ON P.RFCProveedor = PM.RFCProveedor
JOIN Medicina M ON PM.IDInsumoMedicina = M.IDInsumoMedicina
JOIN DistribuirMedicina DM ON M.IDInsumoMedicina = DM.IDInsumoMedicina
JOIN Bioma B ON DM.IDBioma = B.IDBioma
LEFT JOIN Animal A ON B.IDBioma = A.IDBioma
GROUP BY P.RFCProveedor, M.Nombre, B.TipoBioma
HAVING COUNT(A.IDAnimal) > 3;
```

En esta consulta se realizan los juicios necesarios para obtener la información de la medicina que provee un proveedor en un bioma, además, contamos que en el bioma hay más de 3 animales.

	nombre character varying (50)	apellidopaterno character varying (50)	medicinasuministrada character varying (350)	tipobioma character varying (50)	cantidadanimales bigint
1	Chickie	O'Scanlan	Tension Headache Relief	aviario	71
2	Harlene	Jarret	Welchol	tundra	71
3	Reube	Prise	CYPROHEPTADINE HYDROCHLORIDE	bosque tropical	72
4	Tricia	Cocci	Mometasone Furoate	desierto	72
5	Angela	Jenkison	TRAZODONE HYDROCHLORIDE	pastizales	144
6	Lillie	Blencoe	Palmers Amaranth	franja costera	71
7	Micheline	Blunt	Antibacterial Foaming Hand Cleanser	pastizales	72
8	Bobbi	Maharry	Lisinopril	pastizales	72
9	Josh	Sammes	Piroxicam	bosque templado	71
10	Xenos	Aysik	Butorphanol Tartrate	aviario	71
11	Shantee	Masi	Enchanted Moments Berries and Cream Hand Sanitizer	bosque tropical	72

Imagen 1. Resultado de la consulta 1

2. Obtener la cantidad total gastada por cada visitante en tickets, considerando descuentos. Esta consulta suma el costo total de los tickets en la tabla Visitante, y considera si se aplicó algún descuento.

```

SELECT IDVisitante, SUM(CostoUnitario - (CostoUnitario * Descuento / 100)) AS
TotalGastado
FROM Ticket
GROUP BY IDVisitante;

```

En esta consulta solo se realiza una suma del costo unitario - los descuentos que se realizaron en los tickets, después agrupamos por el Id del visitante en orden ascendente por defecto.

	idvisitante integer	totalgastado numeric
1	87	4981.414500
2	652	3698.713800
3	477	2517.163200
4	550	3364.762400
5	51	1033.662500
6	781	6957.146200
7	839	4673.255400
8	858	2975.861800
9	278	4120.353600
10	406	2499.144500

Imagen 2. Resultado de la consulta 2

3. Obtener toda la información del visitante donde se cumpla: la cantidad de servicios vendidos por cada visitante, sin tomar en cuenta los servicios de baño, los visitantes tuvieron que haber comprado este servicio más 2 o más veces. ordenar por fecha de nacimiento de forma ascendente.

```

SELECT *
FROM visitante WHERE idVisitante IN (
    SELECT C.IDVisitante
    FROM Comprar C JOIN Servicio S ON C.IDServicio = S.IDServicio
    WHERE S.TipoServicio <> 'Baño'
    GROUP BY C.IDVisitante, S.TipoServicio
    HAVING COUNT(S.TipoServicio) >= 2)
ORDER BY fechanacimiento ASC;

```

En esta consulta se está realizando una subconsulta: la cual obtiene la información de los servicios que el cliente compró, sin contar el servicio del baño, y además que haya comprado dos o más servicios.

	idvisitante [PK] integer	genero character varying (50)	nombre character varying (50)	paterno character varying (50)	materno character varying (50)	fechanacimiento date
1	108	Hombre	Mellicent	De Bruyne	Lobe	1981-12-05
2	624	No binario	Corney	Spelwood	Aubin	1983-09-04
3	602	No binario	Erick	Starton	Moores	1985-10-13
4	876	Hombre	Son	Lighterness	Tocque	1988-04-14
5	67	Hombre	Oona	Youdell	Sijmons	1990-01-31
6	843	Mujer	Collette	Walliker	Cowitz	1991-10-21
7	913	No binario	Sharleen	Lonsdale	Simkovich	1995-03-06
8	672	Hombre	Liam	Brace	Lathom	1996-02-24
9	994	No binario	Thorn	Endricci	Gulfoyle	1997-04-04
10	471	No binario	Lenora	Gunstone	Crossgrove	1998-01-19
11	873	Hombre	Carlynn	Liddiard	Spurrier	1999-12-21

Imagen 3. Resultado de la consulta 3

4. Seleccionar el tipo de bioma, la especie y contar la cantidad de animales por especie en cada bioma.

```

SELECT b.TipoBioma, a.Especie, COUNT(*) AS CantidadAnimales
FROM Animal a
JOIN Bioma b ON a.IDBioma = b.IDBioma
GROUP BY b.TipoBioma, a.Especie;

```

Esta consulta muestra el tipo de bioma, la especie y la cantidad de animales por especie en cada bioma. Utiliza un JOIN entre las tablas Animal y Bioma para relacionar la información, y la función COUNT con GROUP BY para realizar el conteo por bioma y especie.

	tipobioma character varying (50)	especie character varying (50)	cantidadanimales bigint
1	tundra	Tadorna tadorna	1
2	franja costera	Hystrix cristata	1
3	tundra	Macropus robustus	1
4	franja costera	Procyon lotor	1
5	desierto	Pedetes capensis	1
6	pastizales	Pteropus rufus	1
7	aviario	Ammospermophilus nelsoni	1
8	desierto	Bison bison	1
9	tundra	Panthera leo persica	1
10	aviario	Orcinus orca	1
11	bosque templado	Ratufa indica	1

Imagen 4. Resultado de la consulta 4

5. Seleccionar los 2 eventos con la segunda mayor capacidad en cada tipo de evento

```
WITH RankedEventos AS (
  SELECT e.TipoEvento, e.Capacidad,
         DENSE_RANK() OVER (PARTITION BY e.TipoEvento ORDER BY e.Capacidad DESC) AS
  RankingCapacidad
  FROM Evento e)
SELECT TipoEvento, Capacidad
FROM RankedEventos
WHERE RankingCapacidad = 2 OR RankingCapacidad = 3
ORDER BY TipoEvento, Capacidad DESC;
```

La consulta SQL utiliza una tabla temporal llamada RankedEventos para asignar rangos de capacidad a eventos, divididos por tipo de evento. Luego, selecciona los eventos con el segundo y tercer rango de capacidad en cada tipo de evento. Los resultados finales incluyen el tipo de evento y la capacidad de estos eventos, ordenados por tipo de evento y capacidad descendente. En resumen, la consulta identifica los dos eventos con la segunda mayor capacidad en cada categoría de evento.

	tipoevento character varying (50)	capacidad integer
1	academico	967
2	academico	958
3	dias festivos	930
4	dias festivos	909
5	escolar	945
6	escolar	944
7	infantil	898
8	infantil	868
9	recaudacion de fondos	963
10	recaudacion de fondos	909

Imagen 5. Resultado de la consulta 5

6. Calcular el salario promedio de los cuidadores por especie de animal a cargo

```
SELECT c.Especie, AVG(tc.Salario) AS SalarioPromedio
FROM Cuidar c
JOIN Cuidador tc ON c.RFCCuidador = tc.RFCCuidador
GROUP BY c.Especie;
```

Esta consulta determina el salario promedio de los cuidadores según la especie de animal a su cargo. Utiliza un JOIN entre las tablas Cuidar y Cuidador para relacionar la información, y la función de agregación AVG junto con GROUP BY para calcular el salario promedio por cada especie de animal.

	especie character varying (50)	salariopromedio numeric
1	Raphicerus campestris	1464.00000000
2	Agelaius phoeniceus	14946.00000000
3	Lama pacos	1305.00000000
4	Megaderma spasma	13659.00000000
5	Felis wiedi or Leopardus weidi	1654.00000000
6	Ploceus rubiginosus	99.0000000000
7	Sitta canadensis	6629.66666666
8	Gazella granti	8091.00000000
9	Litrocranius walleri	13857.00000000
10	Eurocephalus anguitimens	5448.00000000

Imagen 6. Resultado de la consulta 6

7. Obtener el tipo de evento y el id de los visitantes de entre 20 y 24 años que han asistido al menos a un evento y la cantidad de visitantes en este rango de edad que asistieron al evento, ordenado de mayor a menor.

```

SELECT tipoevento, count(tipoevento) as cantidad, idvisitante
FROM (SELECT *
      FROM Visitante NATURAL JOIN Visitar
      WHERE AGE(FechaNacimiento) BETWEEN interval '20 years' AND interval '24 years')
NATURAL JOIN evento
GROUP BY tipoevento, idvisitante
ORDER BY tipoevento, cantidad DESC;

```

En esta consulta primero se obtiene a los visitantes que tienen entre 20 y 24 años, utilizando AGE con su fecha de nacimiento. Luego se realiza un natural join con la tabla evento para poder recuperar los eventos a los que asistieron los visitantes en el rango de edad solicitado y así poder obtener cuál es el tipo de evento, el número de asistentes para ese evento (utilizando count) y el identificador del visitante.

	tipoevento character varying (50)	cantidad bigint	idvisitante integer
1	academico	2	397
2	academico	2	754
3	academico	2	9
4	academico	2	521
5	academico	1	279
6	academico	1	616
7	academico	1	395
8	academico	1	399
9	academico	1	413
10	academico	1	286
11	academico	1	708

Imagen 7. Resultado de la consulta 7

8. Obtener la cantidad de tickets de servicios de comida generados por año y por trimestre, ordenados por año y trimestre.

```
SELECT count(numticket), EXTRACT('year' FROM fecha) AS año,
       EXTRACT('quarter' FROM fecha) AS trimestre
FROM ticket
WHERE LOWER(tiposervicio) = 'comida'
GROUP BY año, trimestre
ORDER BY año;
```

Esta consulta se realiza en la tabla Ticket, donde se filtran los tickets en los que se compró el tipo de servicio “comida” y se cuentan cuántos tickets de éste tipo hay, agrupando por año y trimestre, para lo cual se usa un Extract del año y el trimestre usando la fecha que está registrada en el ticket.

	count bigint	año numeric	trimestre numeric
1	14	2021	4
2	20	2022	1
3	11	2022	2
4	17	2022	3
5	13	2022	4
6	14	2023	1
7	22	2023	2
8	10	2023	3
9	23	2023	4
10	17	2024	1
11	17	2024	2

Imagen 8. Resultado de la consulta 8

9. Obtener el directorio (nombre, apellidos, teléfono y correo) de los cuidadores cuyo nombre o apellidos contengan la cadena "or" y que trabajan en el bioma 'aviario'.

```
SELECT nombre, apellidopaterno, apellidomaterno, telefono, correo
FROM (SELECT idbioma, nombre, apellidopaterno, apellidomaterno, telefono, correo
      FROM Cuidador NATURAL JOIN TelefonoCuidador NATURAL JOIN CorreoCuidador
      WHERE (LOWER(nombre) LIKE '%or%' OR LOWER(apellidopaterno) LIKE '%or%'
            OR LOWER(apellidomaterno) LIKE '%or%'))
      NATURAL JOIN Bioma
WHERE LOWER(tipobioma) = 'aviario';
```

En esta consulta primero se realiza un *NATURAL JOIN* entre las tablas Cuidador, TelefonoCuidador y CorreoCuidador para poder recuperar los teléfonos y correos de cada cuidador usando coincidencia por el ID del cuidador. Después se recupera el ID del bioma en que trabaja el cuidador, su nombre, apellido paterno, apellido materno, teléfono y correo de los cuidadores cuyo nombre tiene una subcadena "or", para lo cual se utiliza el operador *LIKE* y luego se realiza un *NATURAL JOIN* con la tabla Bioma para poder obtener ésta información de los cuidadores que trabajan en el bioma 'aviario'. Para que la comparación del tipo de bioma sea insensible a mayúsculas, se utiliza el operador *LOWER*.

	nombre character varying (50)	apellidopaterno character varying (50)	apellidomaterno character varying (50)	telefono character	correo character varying (50)
1	Cortie	Autry	Spurrior	9266957688	edeeganj@wikispaces.com
2	Honor	Sandaver	Darwent	9913224369	cvanbruggen2h@istockphoto.com
3	Aurelie	Sturror	Tollady	1887648420	vmetterick3n@webnode.com
4	Cordie	Scampion	Meakin	9842051812	sgrievson5e@biblegateway.com
5	Rodge	Towe	Whitmore	2488422291	bbilborough5s@phoca.cz
6	Bordie	Grayley	Kearnton	5615092789	jvandendael6r@mayoclinic.com
7	Thorn	Sharrier	Shurey	9773180482	ffuzzard6y@cyberchimps.com
8	Sarena	Galer	Clotworthy	4636365859	bhawney7q@reference.com
9	Iorgo	Allman	De Courtney	9856294192	mhencke7x@patch.com
10	Milena	De Michetti	Grigoryov	8239787252	ctaudevinb1@youtu.be
11	Chloris	Hebden	Blanche	7559725360	rmcivorb8@telegraph.co.uk
12	Shara	Dorken	Ambrosini	5645882815	snanellibf@sbwire.com
13	Pris	Larive	Goford	8712916857	asimonitecs@marriott.com

Imagen 9. Resultado de la consulta 9

10. Encontrar los animales cuyo peso está por encima del promedio de peso de todos los animales en el mismo bioma:

```
WITH PesoPromedioBioma AS (
  SELECT a.IDBioma, AVG(a.Peso) AS PesoPromedio
  FROM Animal a
  GROUP BY a.IDBioma)
```



```

SELECT a.IDAnimal, a.Especie, a.Peso, pb.PesoPromedio AS PesoPromedioBioma
FROM Animal a
JOIN PesoPromedioBioma pb ON a.IDBioma = pb.IDBioma
WHERE a.Peso > pb.PesoPromedio;

```

La consulta encuentra y muestra los animales que tienen un peso superior al promedio de peso de todos los animales en el mismo bioma. Cada fila en los resultados representa un animal con su especie, peso y el promedio de peso de su bioma.

	idanimal [PK] integer	especie character varying (50)	peso numeric	pesopromediobioma numeric
1	4	Procyon lotor	271.71	151.3123943661971831
2	8	Lamprotornis sp.	168.83	145.1862500000000000
3	9	Boa constrictor mexicana	233.98	154.3168055555555556
4	10	Nycticorax nycticorax	190.68	155.8319444444444444
5	11	Crotalus adamanteus	266.39	151.3123943661971831
6	13	Ammospermophilus nelsoni	151.75	144.9661971830985915
7	14	Anas punctata	294.0	164.5639436619718310
8	16	Chauna torquata	285.51	154.3168055555555556
9	17	Ephippiorhynchus senegalen...	296.63	155.8319444444444444
10	18	Platalea leucordia	294.88	151.3123943661971831
11	19	Canis lupus baileyi	279.58	150.0732394366197183

Imagen 10. Resultado de la consulta 10

11. Obtener el salario promedio de los veterinarios por especialidad:

```

SELECT v.Especialidad, AVG(v.Salario) AS SalarioPromedio
FROM Veterinario v
GROUP BY v.Especialidad;

```

La consulta busca información en una tabla llamada "Veterinario". Agrupa a los veterinarios según su especialidad y calcula el salario promedio para cada grupo de especialidad. Entonces, en lugar de tener una fila para cada veterinario, obtienes una fila para cada especialidad con el salario promedio asociado a esa especialidad. En resumen, la consulta devuelve las especialidades de los veterinarios y el salario promedio para cada una.

	especialidad character varying (50)	salariopromedio numeric
1	Oftalmología	6226.420000000
2	Nutrición	3270.985000000
3	Rehabilitación	4086.716666666
4	Endocrinología	5275.185000000
5	Cirugía	3614.972000000
6	Cardiología	7629.287500000
7	Dermatología	4193.437500000
8	Odontología	566.202500000
9	Oncología	2929.943333333
10	Neurología	3530.110000000

Imagen 11. Resultado de la consulta 11

12. Encontrar los cuidadores cuyo salario sea mayor que el salario promedio de todos los cuidadores:

```
WITH SalarioPromedioCuidador AS (
  SELECT AVG(Salario) AS PromedioSalarioCuidador
  FROM Cuidador)
SELECT RFCuidador, Nombre, ApellidoPaterno, ApellidoMaterno, Salario
FROM Cuidador
WHERE Salario > (SELECT PromedioSalarioCuidador FROM SalarioPromedioCuidador);
```

La consulta encuentra y muestra los detalles de los cuidadores que tienen salarios más altos que el promedio de salarios de todos los cuidadores. Cada fila en los resultados representa a un cuidador con su nombre, apellido y salario, destacando aquellos que ganan por encima del promedio.

	rfccuidador [PK] character varying (13)	nombre character varying (50)	apellidopaterno character varying (50)	apellidomaterno character varying (50)	salario numeric
1	LBLD38126934X	Vinson	Dow	Robilliard	7588
2	WRVR032646V9V	Murdoch	Kehoe	Stickler	8885
3	ZCNE630985LGH	Gertruda	Durward	Soppett	8990
4	CHYI217969PDO	Cyril	Yetton	Neeson	7118
5	TWZT232750IVI	Clarette	aManger	Perot	9637
6	KIFX045181Q9N	Alikee	Wills	Freyne	8873
7	MK0X4769199EU	Staffard	Scrogges	Jordison	14363
8	VSXG586833OYW	Vivyan	Carleman	Bertwistle	7158
9	AMAN215457U5A	Corena	Medway	Haccleton	14934
10	HLRW6343963MF	Morty	Garnall	Halewood	14406

Imagen 12. Resultado de la consulta 12

13. Obtener la tabla de cada animal con uno de los cuidadores y su veterinario. Ordenado por idAnimal. Si existe más de un cuidador, tomar sólo uno.

```
SELECT a.IDAnimal, a.Especie, a.NombreAnimal, a.IDBioma, (
    SELECT MAX(c.RFCCuidador)
    FROM Cuidar c
    WHERE c.IDAnimal = a.IDAnimal
) AS RFCCuidador, v.RFCVeterinario
FROM Animal a
JOIN Atender v ON a.IDAnimal = v.IDAnimal
ORDER BY idanimal;
```

En esta consulta primero se calcula el máximo del RFCCuidador, esto no porque nos interesa el máximo, más bien para conservar solamente un Cuidador de cada animal, ya que pueden haber más de uno. Después se hace un JOIN con la tabla Atender, esto para agregar la columna del veterinario por cada animal.

	idanimal integer	especie character varying (50)	nombreal character varying (50)	idbioma integer	rfccuidador text	rfcveterinario character varying (13)
1	1	Microcebus murinus	"Lemur, grey mouse"	1	LBLD38126934X	GKBI031660A54
2	2	Morelia spilotes variegata	"Snake, carpet"	2	WRVR032646V9V	DCPL304984QOL
3	3	Libellula quadrimaculata	"Dragonfly, russian"	3	XCNJ255417P5G	PGCC54126860L
4	4	Procyon lotor	Common raccoon	4	ZCNE630985LGH	APTX9931192LJ
5	5	Theropithecus gelada	"Monkey, bleeding heart"	5	CHYI217969PDO	FZTX441413ZS8
6	6	Mycteria ibis	Yellow-billed stork	6	KFYW89384513J	GZNB080437AXT
7	7	Zalophus californicus	California sea lion	7	JDWA625854R18	AATT966722NZG
8	8	Lamprotornis sp.	Glossy starling (unidentified)	1	TWZT232750IVI	DGVO091266ZSW
9	9	Boa constrictor mexicana	Mexican boa	2	KIFX045181Q9N	TBUJ5623783EG
10	10	Nycticorax nycticorax	"Heron, black-crowned night"	3	LIQF318191FE9	OGZR6838126UF

Imagen 13. Resultado de la consulta 13

14. Obtener las jaulas donde el animal más alto sea una hembra. Y mostrar el animal más alto de cada jaula.

```
SELECT a.IDJaula, a.IDAnimal, a.NombreAnimal, a.Altura, a.Sexo
FROM Animal a
JOIN (
    SELECT IDJaula, MAX(Altura) AS AlturaMaxima
    FROM Animal
    GROUP BY IDJaula
) AS AlturaMaximaJaula ON a.IDJaula = AlturaMaximaJaula.IDJaula AND a.Altura =
AlturaMaximaJaula.AlturaMaxima
WHERE a.Sexo = 'Hembra';
```

En esta consulta primero se calcula la altura máxima de entre todos los animales de cada jaula. Después se hace un JOIN para obtener el animal de altura máxima de cada jaula, que además es hembra.

	idjaula integer	idanimal [PK] integer	nombreakanimal character varying (50)	altura numeric	sexo character varying (50)
1	184	185	"Cow, scottish highland"	4.94	Hembra
2	87	88	"Boa, columbian rainbow"	2.85	Hembra
3	273	274	"Bee-eater, carmine"	3.13	Hembra
4	51	52	"Turaco, violet-crested"	4.78	Hembra
5	190	191	"Hornbill, yellow-billed"	3.46	Hembra
6	350	351	Common wombat	2.98	Hembra
7	278	279	Jungle cat	4.11	Hembra
8	424	425	"Oryx, beisa"	2.7	Hembra
9	406	407	"Oryx, beisa"	1.97	Hembra
10	292	293	"Siskin, yellow-rumped"	3.07	Hembra
11	271	272	Sloth bear	1.06	Hembra

Imagen 14. Resultado de la consulta 14

15. Proveedores que proveen medicina y alimento, que después se distribuye al mismo bioma.

```

SELECT DISTINCT p.RFCProveedor, p.Nombre, p.ApellidoPaterno, p.ApellidoMaterno,
b.IDBioma
FROM Proveedor p
JOIN ProveerMedicina pm ON p.RFCProveedor = pm.RFCProveedor
JOIN Medicina m ON pm.IDInsumoMedicina = m.IDInsumoMedicina
JOIN DistribuirMedicina dm ON m.IDInsumoMedicina = dm.IDInsumoMedicina
JOIN Bioma b ON dm.IDBioma = b.IDBioma
JOIN ProveerAlimento pa ON p.RFCProveedor = pa.RFCProveedor
JOIN Alimento a ON pa.IDInsumoAlimento = a.IDInsumoAlimento
JOIN DistribuirAlimento da ON a.IDInsumoAlimento = da.IDInsumoAlimento
WHERE dm.IDBioma = da.IDBioma;

```

En esta consulta se selecciona al proveedor, luego se hacen múltiples JOIN, para obtener una tabla con la información de la Medicina que provee y del bioma al cual es distribuido. Posteriormente se hace un JOIN con el mismo proceso pero ahora con Alimento en donde el bioma es el mismo. Y se utiliza DISTINCT para evitar cuando hay más de un alimento o medicina que es distribuido al mismo bioma, lo cual generaría filas iguales.

	rfcproveedor character varying (13) 	nombre character varying (50) 	apellidopaterno character varying (50) 	apellidomaterno character varying (50) 	idbioma integer 
1	ABKR748220D0V	Jefferson	Fardy	Artrick	7
2	ADIX931397A1I	Edwina	Willcot	Spracklin	7
3	AZGR599270UEW	Rudyard	Tabourin	Meddings	2
4	BJRH399376L3F	Pearl	Badger	Durman	5
5	BKSV3451673ID	Alastair	Sponer	Asken	4
6	BUHO450272J9S	Karol	Meynell	Sollom	2
7	BZII060616Z0B	Valeda	Vernazza	Bahde	2
8	CBIH511726WL8	Alicea	Sherlock	Billingsly	7
9	CHCW316779HF0	Kandace	Askaw	Clatworthy	1
10	CJCY554950MHJ	Hamel	Lecointe	Devita	5

Imagen 15. Resultado de la consulta 15