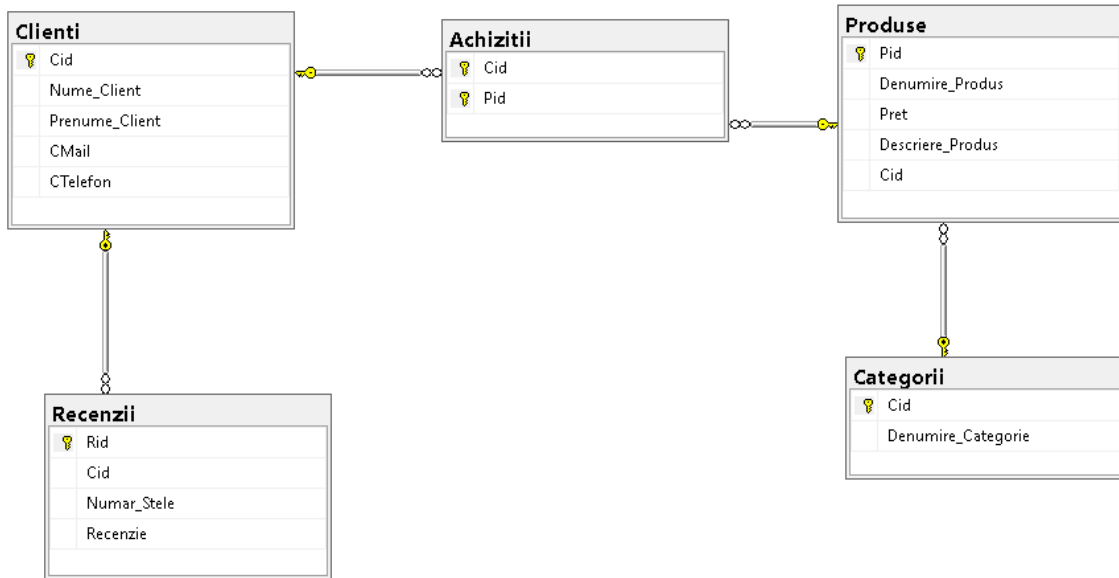## Queries- Examples

We consider the following database:


```sql
CREATE DATABASE FotoShop
GO
USE FotoShop
GO

CREATE TABLE Categorii(
Cid INT PRIMARY KEY IDENTITY,
Denumire_Categorie VARCHAR(30))

CREATE TABLE Produse(
Pid INT PRIMARY KEY IDENTITY,
Denumire_Produs VARCHAR(100),
Pret INT CHECK(Pret>0) NOT NULL,
Descriere_Produs VARCHAR(1000),
Cid INT FOREIGN KEY REFERENCES
Categorii(Cid))

CREATE TABLE Clienti(
Cid INT PRIMARY KEY IDENTITY,
Nume_Client VARCHAR(50),
Prenume_Client VARCHAR(50),
CMail VARCHAR(50),
CTelefon VARCHAR(50)
)
```

```sql
CREATE TABLE Achizitii(
Cid INT FOREIGN KEY REFERENCES
Clienti(Cid),
Pid INT FOREIGN KEY REFERENCES Produse(Pid)
CONSTRAINT Pk_Achizitii PRIMARY
KEY(Cid,Pid)
)

CREATE TABLE Recenzii(
Rid int primary key identity,
Cid INT FOREIGN KEY REFERENCES
Clienti(Cid),
Numar_Stele INT CHECK(Numar_Stele>0 AND
Numar_Stele<=5) NOT NULL,
Recenzie VARCHAR(1000)
)
```

```sql
USE FotoShop
GO

-- Categorii
INSERT INTO Categorii(Denumire_Categorie) VALUES ('Profesional'), ('Ocazional'), ('Uzual')
```

```
-- Produse
-- delete from Produse
INSERT INTO Produse(Denumire_Produs, Pret, Descriere_Produs, Cid)
VALUES ('Canon d5K','8800','Aparat de fotografiat Canon',2)
INSERT INTO Produse(Denumire_Produs, Pret, Descriere_Produs, Cid)
VALUES ('Nikon D850','10000','Aparat de fotografiat Nikon',3)
INSERT INTO Produse(Denumire_Produs, Pret, Descriere_Produs, Cid)
VALUES ('Sony Alpha9','25000','Aparat de fotografiat Sony',2)

-- Clienti
INSERT INTO Clienti(Nume_Client, Prenume_client, CMail, CTelefon)
VALUES ('Almasan','Radu','a@acd.ro','0752525522'), ('Cristea', 'Docolin', 'sd@gfby.com',
'0712121212')

-- Achizitii
Insert into Achizitii(Cid, Pid) Values (1, 4), (1, 5)

-- Recenzii
insert into Recenzii(Cid, Numar_Stele, Recenzie) Values (1, 4, 'bun'), (2, 5, 'FB')
```

```
-- select
select * from Categorii
select * from Produse
select * from Clienti
select * from Achizitii
select * from Recenzii
```

Cid	Denumire_Categorie
1	Profesional
2	Ocazional
3	Uzual

Pid	Denumire_Produs	Pret	Descriere_Produs	Cid
4	Canon d5K	8800	Aparat de fotografiat Canon	2
5	Nikon D850	10000	Aparat de fotografiat Nikon	3
6	Sony Alpha9	25000	Aparat de fotografiat Sony	2

Cid	Nume_Client	Prenume_Client	CMail	CTelefon
1	Almasan	Radu	a@acd.ro	0752525522
2	Cristea	Docolin	sd@gfby.com	0712121212

Cid	Pid
1	4
1	5

Rid	Cid	Numar_Stele	Recenzie
1	1	4	bun
2	2	5	FB

```
SELECT fields_name separated with comma (,)
FROM tables_name
WHERE conditions_related_to_the_fields_from_select
GROUP BY fileds_name_the_same_as_in_select
HAVING aggregate_funtion_condition (avg, sum, min, max, count)
ORDER BY field(s)_name
```

SELECT WITH *WHERE*

```
SELECT Pid, Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs = 'Canon d5K' OR Pid>=4
```

Pid	Denumire_Produs	Pret
4	Canon d5K	8800
5	Nikon D850	10000
6	Sony Alpha9	25000

```sql
SELECT Pid, Denumire_Produs, Descriere_Produs
FROM Produse
WHERE Denumire_Produs LIKE '%Alpha%' AND
Descriere_Produs LIKE 'Aparat%'
-- contine Alpha, incepe cu Aparat
```

	Pid	Denumire_Produs	Descriere_Produs
1	6	Sony Alpha9	Aparat de fotografiat Sony

```sql
SELECT Pid, Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs = 'C_%' OR Pret BETWEEN 500
AND 10000
-- incepe cu C si are cel putin 2 caractere
```

	Pid	Denumire_Produs	Pret
1	4	Canon d5K	8800
2	5	Nikon D850	10000

```sql
SELECT Pid, Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs IN ('Canon', 'Nikon', 'Sony')
```

	Pid	Denumire_Produs	Pret

SELECT WITH *ORDER BY*

```sql
SELECT Pid, Denumire_Produs, Pret
FROM Produse
ORDER BY Denumire_Produs ASC
-- equivalent
SELECT Pid, Denumire_Produs, Pret
FROM Produse
ORDER BY Denumire_Produs
```

	Pid	Denumire_Produs	Pret
1	4	Canon d5K	8800
2	5	Nikon D850	10000
3	6	Sony Alpha9	25000

	Pid	Denumire_Produs	Pret
1	4	Canon d5K	8800
2	5	Nikon D850	10000
3	6	Sony Alpha9	25000

```sql
SELECT Pid, Denumire_Produs, Pret
FROM Produse
WHERE Pret >500
ORDER BY Pret DESC
```

	Pid	Denumire_Produs	Pret
1	6	Sony Alpha9	25000
2	5	Nikon D850	10000
3	4	Canon d5K	8800

```sql
SELECT Pid, Denumire_Produs, Pret
FROM Produse
WHERE Pret BETWEEN 8000 AND 10000
ORDER BY Pret DESC, Denumire_Produs
```

	Pid	Denumire_Produs	Pret
1	5	Nikon D850	10000
2	4	Canon d5K	8800

Distinct, Top, arithmetical expressions

Distinct

```sql
insert into Produse(Denumire_Produs, Pret, Descriere_Produs, Cid)
VALUES ('Canon', 8800, 'Aparat Canon', 1)

-- Distinct
SELECT Pret
FROM Produse

SELECT DISTINCT Pret
FROM Produse
```

	Pret
1	8800
2	10000
3	25000
4	8800

	Pret
1	8800
2	10000
3	25000

Top

``sql
SELECT Top 2 Pret
FROM Produse
``
	Results | Messages
	
	   Pret
	1 8800
	2 10000

Arithmetical expressions

``sql
SELECT Denumire_Produs, Pret, Pret-100 as
Pret_nou, Pret_vechi=Pret*2- Pret/2, Pret/5
FROM Produse
``
	Results | Messages
	
	  Denumire_Produs Pret Pret_nou Pret_vechi (No column name)
	1 Canon d5K 8800 8700 13200 1760
	2 Nikon D850 10000 9900 15000 2000
	3 Sony Alpha9 25000 24900 37500 5000
	4 Canon 8800 8700 13200 1760

SELECT WITH **GROUP BY**

``sql
-- select with GROUP BY
 	
SELECT AVG(Pret) AS Pret_mediu
FROM Produse
 	
select MAX(Pret)
FROM Produse
``
	Results | Messages
	
	  Pret_mediu
	1 13150
	
	  (No column name)
	1 25000

``sql
SELECT Pid, AVG(Pret) AS PretMediu, SUM(Pret) AS
PretTotal
FROM Produse
GROUP BY Pid
 	
SELECT Pid, Denumire_Produs, AVG(Pret) AS PretMediu,
SUM(Pret) AS PretTotal
FROM Produse
GROUP BY Pid, Denumire_Produs
ORDER BY Pid
``
	Results | Messages
	
	  Pid PretMediu PretTotal
	1 4 8800 8800
	2 5 10000 10000
	3 6 25000 25000
	4 7 8800 8800
	
	  Pid Denumire_Produs PretMediu PretTotal
	1 4 Canon d5K 8800 8800
	2 5 Nikon D850 10000 10000
	3 6 Sony Alpha9 25000 25000
	4 7 Canon 8800 8800

``sql
SELECT Pid, Denumire_Produs, AVG(Pret) AS PretMediu,
SUM(Pret) AS PretTotal
FROM Produse
GROUP BY Pid
-- error - the same number of columns + the same columns
from SELECT also in GROUP BY
``
	Msg 8120, Level 16, State 1, Line 214
	Column 'Produse.Denumire_Produs' is
	invalid in the select list because it
	is not contained in either an
	aggregate function or the GROUP BY
	clause.

``sql
SELECT Cid, AVG(Pret) AS Pret_mediu
FROM Produse
WHERE Pret>9000 -- condition(s) for the fields from the
table(s)
GROUP BY Cid
HAVING AVG(Pret)>20000 OR SUM(Pret) BETWEEN 12000 AND
1200000 -- Condition(s) for the aggregate functions
ORDER BY Cid
``
	Results | Messages
	
	  Cid Pret_mediu
	1 2 25000

`-- error`	Msg 156, Level 15, State 1, Line 232

| ```sql
SELECT Cid, AVG(Pret) AS Pret_mediu
FROM Produse
WHERE Pret>9000
HAVING AVG(Pret)>2000
GROUP BY Cid
``` | Incorrect syntax near the keyword 'GROUP'. |
| ```sql
-- error
SELECT Cid, AVG(Pret) AS Pret_mediu
FROM Produse
WHERE Pret>9000 AND AVG(Pret)>20000
GROUP BY Cid
``` | Msg 147, Level 15, State 1, Line 237 An aggregate may not appear in the WHERE clause unless it is in a subquery contained in a HAVING clause or a select list, and the column being aggregated is an outer reference. |
| ```sql
-- error
SELECT Cid, AVG(Pret) AS Pret_mediu
FROM Produse
GROUP BY Cid
HAVING AVG(Pret)>20000 OR SUM(Pret) BETWEEN 12000 AND
1200000
WHERE Pret > 10000
``` | Msg 156, Level 15, State 1, Line 245 Incorrect syntax near the keyword 'WHERE'. |

SELECT IN SELECT (Nested queries)

| ```sql
-- SELECT IN SELECT (Nested queries)

SELECT p.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c INNER JOIN Produse p ON c.Cid=p.Cid
WHERE Pret>5000

-- equivalent IN
SELECT p.Cid, p.Denumire_Produs, p.Pret
FROM Produse p
WHERE p.Pret>5000 and p.Cid IN (SELECT c.Cid FROM Categorii c)

-- equivalent EXISTS
SELECT p.Cid, p.Denumire_Produs, p.Pret
FROM Produse p
WHERE p.Pret>5000 and EXISTS (SELECT * FROM Categorii c WHERE c.Cid=p.Cid)

-- equivalent FROM
SELECT A.Cid, A.Denumire_Produs, A. Pret
FROM (SELECT p.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c INNER JOIN Produse p ON c.Cid=p.Cid
WHERE Pret>5000) A
``` | Results   Messages

	Cid	Denumire_Produs	Pret
1	2	Canon d5K	8800
2	3	Nikon D850	10000
3	2	Sony Alpha9	25000
4	1	Canon	8800

SELECT INTO

| ```sql
IF OBJECT_ID ('dbo.TabelInserare', 'U') IS NOT
NULL
DROP TABLE dbo.TabelInserare;
GO
-- Create TabelInserare
SELECT Denumire_Produs, Pret
INTO dbo.TabelInserare
FROM Produse
``` | (2 row(s) affected)

Results   Messages

	Denumire_Produs	Pret
1	Canon d5K	8800
2	Canon	8800

```
WHERE Denumire_Produs LIKE 'Canon%'


select * from TabelInserare
```

SELECT WITH **INNER JOIN**

```
select * from Produse
select * from Categorii
```

	Pid	Denumire_Produs	Pret	Descriere_Produs	Cid
1	4	Canon d5K	8800	Aparat de fotografiat Canon	2
2	5	Nikon D850	10000	Aparat de fotografiat Nikon	3
3	6	Sony Alpha9	25000	Aparat de fotografiat Sony	2
4	7	Canon	8800	Aparat Canon	1
5	8	Canon v	4000	Aparat de fotografiat Canon	NULL

	Cid	Denumire_Categorie
1	1	Profesional
2	2	Ocazional
3	3	Uzual
4	4	test
5	5	test1

```
-- without inner join – will be all the
combinations, but not all of them exists
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c, Produse p

-- with WHERE and equivalent with INNER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c, Produse p
WHERE c.Cid=p.Cid

-- INNER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c INNER JOIN Produse p ON
c.Cid=p.Cid
```

	Cid	Denumire_Produs	Pret
1	1	Canon d5K	8800
2	2	Canon d5K	8800
3	3	Canon d5K	8800
4	4	Canon d5K	8800
5	5	Canon d5K	8800
6	1	Nikon D850	10000
7	2	Nikon D850	10000
8	3	Nikon D850	10000

	Cid	Denumire_Produs	Pret
1	2	Canon d5K	8800
2	3	Nikon D850	10000
3	2	Sony Alpha9	25000
4	1	Canon	8800

	Cid	Denumire_Produs	Pret
1	2	Canon d5K	8800
2	3	Nikon D850	10000
3	2	Sony Alpha9	25000
4	1	Canon	8800

INNER JOIN

```
-- INNER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c INNER JOIN Produse p ON c.Cid=p.Cid
WHERE Pret>5000
-- all the values that communicate from both tables
(Produse, Categorii) - NO NULL values
```

	Cid	Denumire_Produs	Pret
1	2	Canon d5K	8800
2	3	Nikon D850	10000
3	2	Sony Alpha9	25000
4	1	Canon	8800

LEFT OUTER JOIN

```
-- LEFT OUTER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c LEFT OUTER JOIN Produse p ON c.Cid=p.Cid
-- all from Categorii (the table from left side), even if
there are no records coresponding in Produse (NULL value)
```

	Cid	Denumire_Produs	Pret
1	1	Canon	8800
2	2	Canon d5K	8800
3	2	Sony Alpha9	25000
4	3	Nikon D850	10000
5	4	NULL	NULL
6	5	NULL	NULL

RIGHT OUTER JOIN

```
-- RIGHT OUTER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c RIGHT OUTER JOIN Produse p ON c.Cid=p.Cid
-- all from Produse (the table from the right side), even
if there are no records coresponding in Categorii (NULL
value)
```

	Cid	Denumire_Produs	Pret
1	2	Canon d5K	8800
2	3	Nikon D850	10000
3	2	Sony Alpha9	25000
4	1	Canon	8800
5	NULL	Canon v	4000

FULL OUTER JOIN

```
-- FULL OUTER JOIN
SELECT c.Cid, p.Denumire_Produs, p.Pret
FROM Categorii c FULL OUTER JOIN Produse p ON c.Cid=p.Cid
-- all from Categorii AND Produse (NULL values from left and
right side) - LEFT OUTER JOIN+ RIGHT OUTER JOIN
```

	Cid	Denumire_Produs	Pret
1	1	Canon	8800
2	2	Canon d5K	8800
3	2	Sony Alpha9	25000
4	3	Nikon D850	10000
5	4	NULL	NULL
6	5	NULL	NULL
7	NULL	Canon v	4000

m-n relationship

```
-- join for m-n relationship
SELECT *
FROM Produse p INNER JOIN Achizitii a on p.Pid=a.Pid
INNER JOIN Clienti c on c.Cid=a.Cid
```

	Pid	Denumire_Produs	Pret	Descriere_Produs	Cid	Cid	Pid	Cid	Nume_Client	Prenume_Client	CMail	CTelefon
1	4	Canon d5K	8800	Aparat de fotografiat Canon	2	1	4	1	Almasan	Radu	a@acd.ro	0752525522
2	5	Nikon D850	10000	Aparat de fotografiat Nikon	3	1	5	1	Almasan	Radu	a@acd.ro	0752525522

CROSS JOIN

```
-- cross join – the Cartesian product = all the
possible combinations with the selected fields
select P.Pid, C.Cid
from Produse P CROSS JOIN Categorii C
```

OR

```
IF OBJECT_ID ('dbo.CrossJ', 'U') IS NOT NULL
DROP TABLE dbo.CrossJ;
GO
-- Create CrossJ
SELECT P.Pid, C.Cid
INTO dbo.CrossJ
from Produse P CROSS JOIN Categorii C
-- select from the new created table
select * from CrossJ
```

	Pid	Cid
1	4	1
2	5	1
3	6	1
4	7	1
5	8	1
6	4	2
7	5	2
8	6	2
9	7	2
10	8	2
11	4	3
12	5	3
13	6	3
14	7	3
15	8	3
16	4	4
17	5	4
18	6	4
19	7	4
20	8	4
21	4	5
22	5	5
23	6	5
24	7	5
25	8	5

Query executed successfully. (25 row(s) affected)

SELECT WITH **UNION, INTERSECT, EXCEPT**

```
select * from Produse
```

	Pid	Denumire_Produs	Pret	Descriere_Produs	Cid
1	4	Canon d5K	8800	Aparat de fotografiat Canon	2
2	5	Nikon D850	10000	Aparat de fotografiat Nikon	3
3	6	Sony Alpha9	25000	Aparat de fotografiat Sony	2
4	7	Canon	8800	Aparat Canon	1
5	8	Canon v	4000	Aparat de fotografiat Canon	NULL

UNION

```
-- UNION - produsele ce au pretul >10000 SAU denumirea incepe cu C
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Pret>10000
UNION
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs LIKE 'C%'
```

	Denumire_Produs	Pret
1	Canon	8800
2	Canon d5K	8800
3	Canon v	4000
4	Sony Alpha9	25000

```
-- UNION ALL - also the duplicates
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Pret>10000
UNION ALL
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs LIKE 'C%'
```

	Denumire_Produs	Pret
1	Sony Alpha9	25000
2	Canon d5K	8800
3	Canon	8800
4	Canon v	4000

INTERSECT

```
-- INTERSECT - produsele ce au pretul >10000 SI denumirea incepe cu C
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Pret>10000
INTERSECT
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs LIKE 'C%'
```

EXCEPT


```
-- EXCEPT - produsele ce au pretul >10000 DAR denumirea NU incepe cu C
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Pret>10000
EXCEPT
SELECT Denumire_Produs, Pret
FROM Produse
WHERE Denumire_Produs LIKE 'C%'
```

`-- for all -- error`
`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Pret>10000`
`ORDER BY Denumire_Produs`
`UNION`
`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Denumire_Produs LIKE 'C%'`
	`Msg 156, Level 15, State 1, Line 360`
	`Incorrect syntax near the keyword 'UNION'.`

`-- order by can be used ONLY at the end, after the both queries were performed`
`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Pret>10000`
`UNION`
`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Denumire_Produs LIKE 'C%'`
`ORDER BY Denumire_Produs`	

`-- the fields from the both SELECT's should have the same number, type and order`
`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Pret>10000`
`UNION`
`SELECT Denumire_Produs`
`FROM Produse`
`WHERE Denumire_Produs LIKE 'C%'`
	`Msg 205, Level 16, State 1, Line 385`
	`All queries combined using a UNION, INTERSECT or EXCEPT operator must have an equal number of expressions in their target lists.`

`SELECT Denumire_Produs, Pret`
`FROM Produse`
`WHERE Pret>10000`
`UNION`
`SELECT Pret, Denumire_Produs`
`FROM Produse`
`WHERE Denumire_Produs LIKE 'C%'`
	`Msg 245, Level 16, State 1, Line 393`
	`Conversion failed when converting the varchar value 'Sony Alpha9' to data type int.`