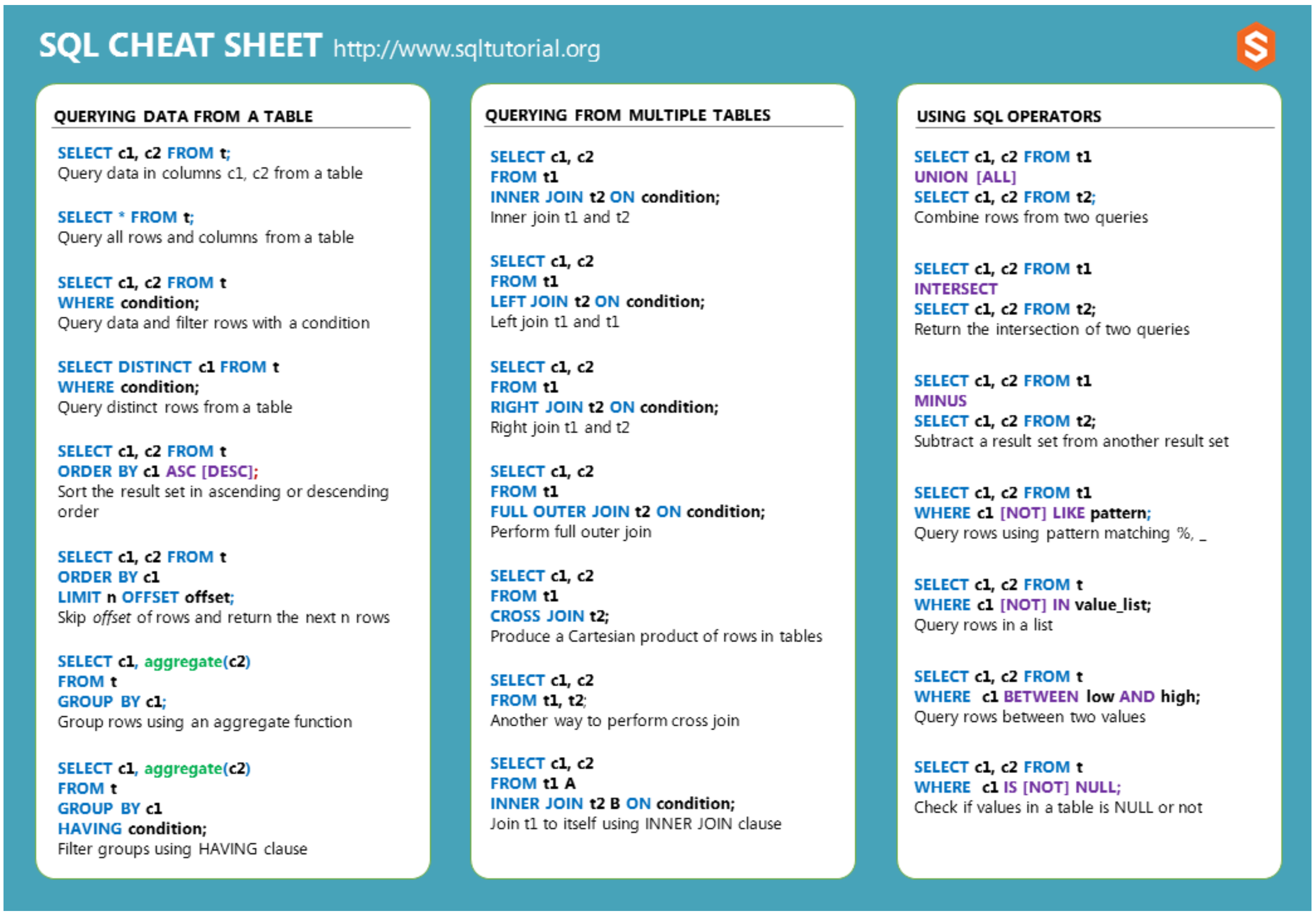
SQL



SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

Import Database

```
connction->bottom left -> administrator -> Data import/Restore
```

Export Database

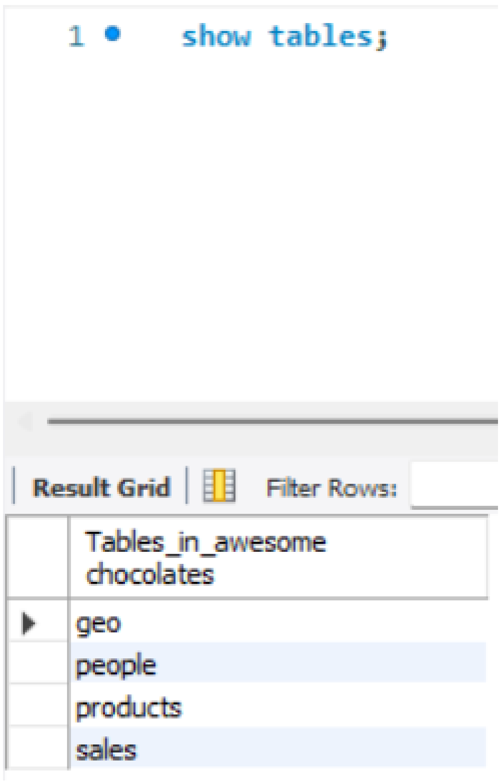
```
connction->bottom left -> administrator -> Data Export
```

Basic

- Create a new sql query tab.
- Double click on database name to select it.
- Strings are wrapped inside of singlequote `(' ')`.
- And Or operator are used to combine multiple condition.
- Not equal is represented by `(<>)`.

List all the tables

```sql
%%Example / Syntax%%
    show tables;
```



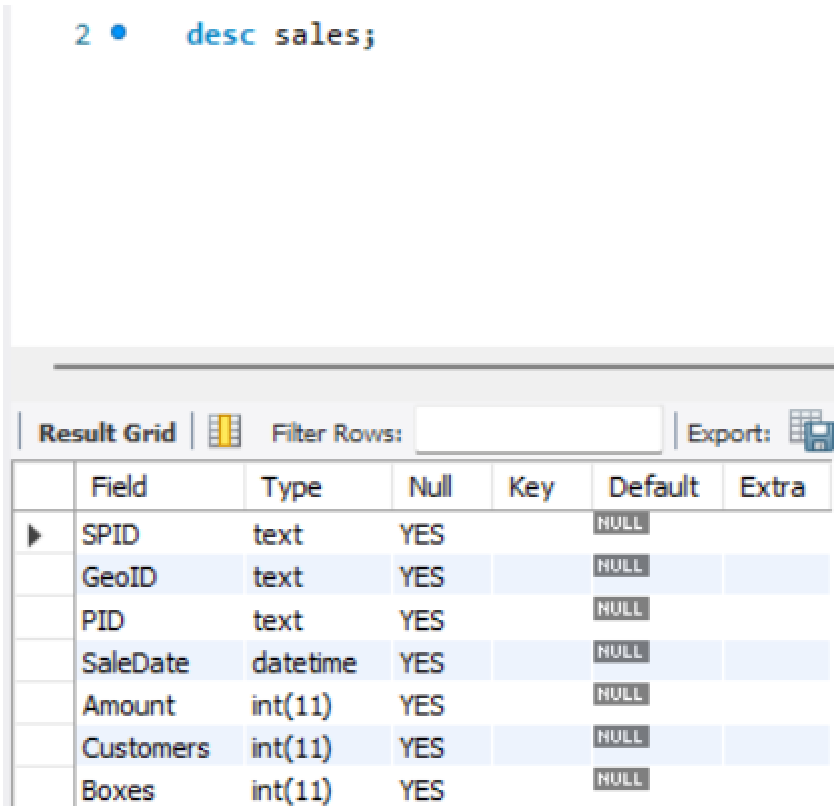
```
show tables;
```

	Tables_in_awesome chocolates
▶	geo
	people
	products
	sales

To describe what are available inside the table

```sql
%%Syntax%%
    desc table_name;
```

```sql
%%Example%%
    desc sales;
```



```sql
desc sales;
```

Result Grid | Filter Rows: | Export:

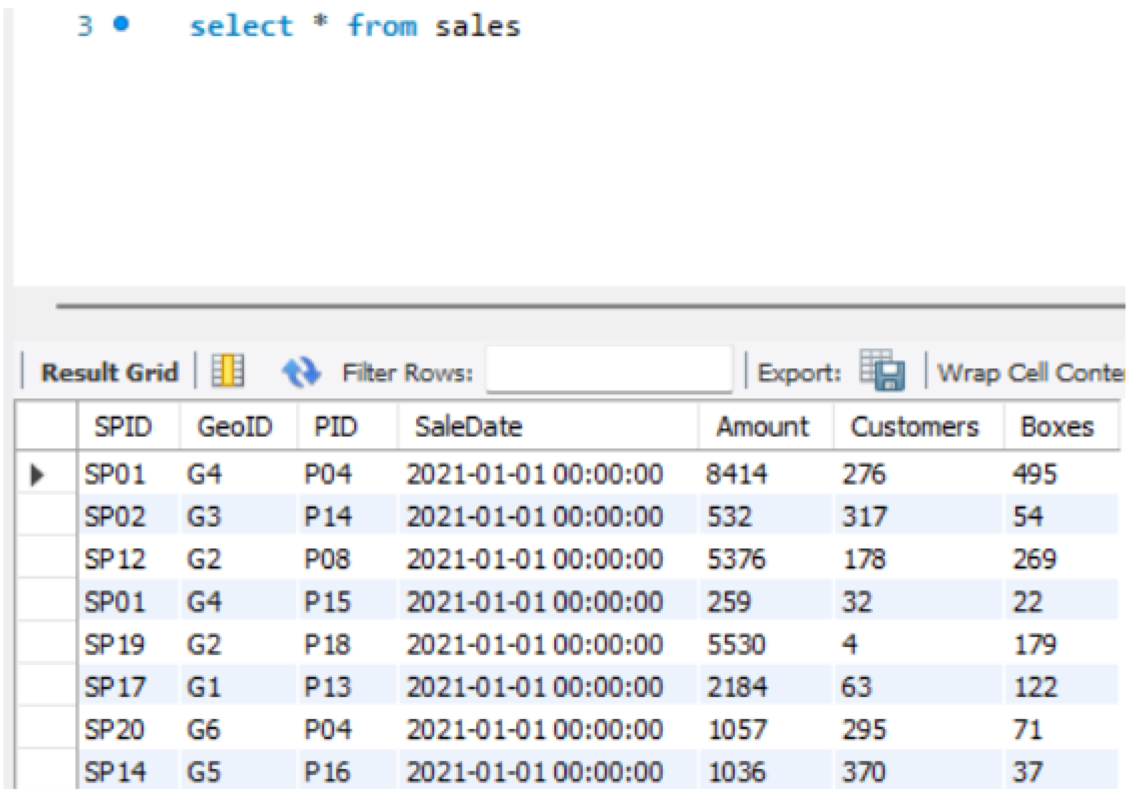
Field	Type	Null	Key	Default	Extra
SPID	text	YES		NULL	
GeoID	text	YES		NULL	
PID	text	YES		NULL	
SaleDate	datetime	YES		NULL	
Amount	int(11)	YES		NULL	
Customers	int(11)	YES		NULL	
Boxes	int(11)	YES		NULL	

Fetching data

`SELECT` statement is used to retrieve data stored in our tables.

```SQL
%%Syntax%%
    select * from table_name;
```

```SQL
%%Example%%
    select * from sales;
```

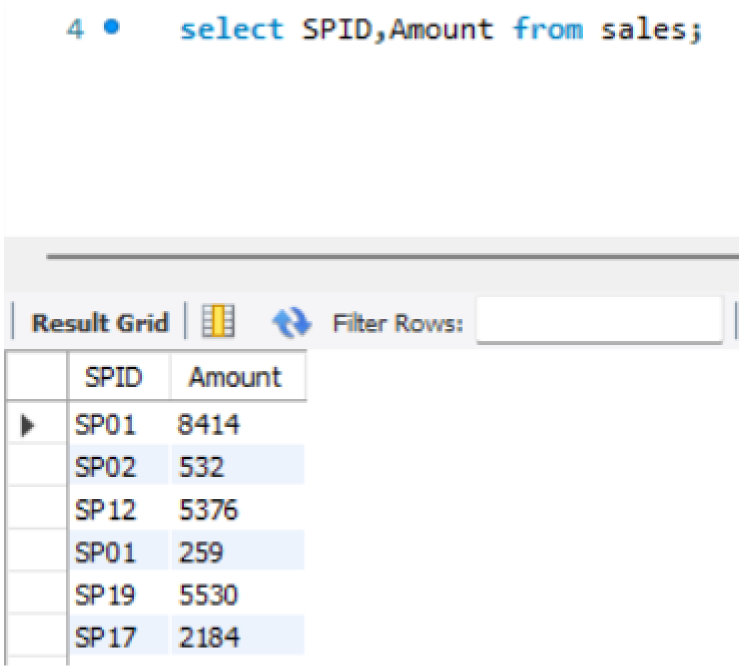


```sql
select * from sales
```

SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP01	G4	P04	2021-01-01 00:00:00	8414	276	495
SP02	G3	P14	2021-01-01 00:00:00	532	317	54
SP12	G2	P08	2021-01-01 00:00:00	5376	178	269
SP01	G4	P15	2021-01-01 00:00:00	259	32	22
SP19	G2	P18	2021-01-01 00:00:00	5530	4	179
SP17	G1	P13	2021-01-01 00:00:00	2184	63	122
SP20	G6	P04	2021-01-01 00:00:00	1057	295	71
SP14	G5	P16	2021-01-01 00:00:00	1036	370	37

```sql
%%Syntax%%
    select col_1_name, col_2_name from table_name;
```

```sql
%%Example%%
    select SPID, Amount from sales;
```



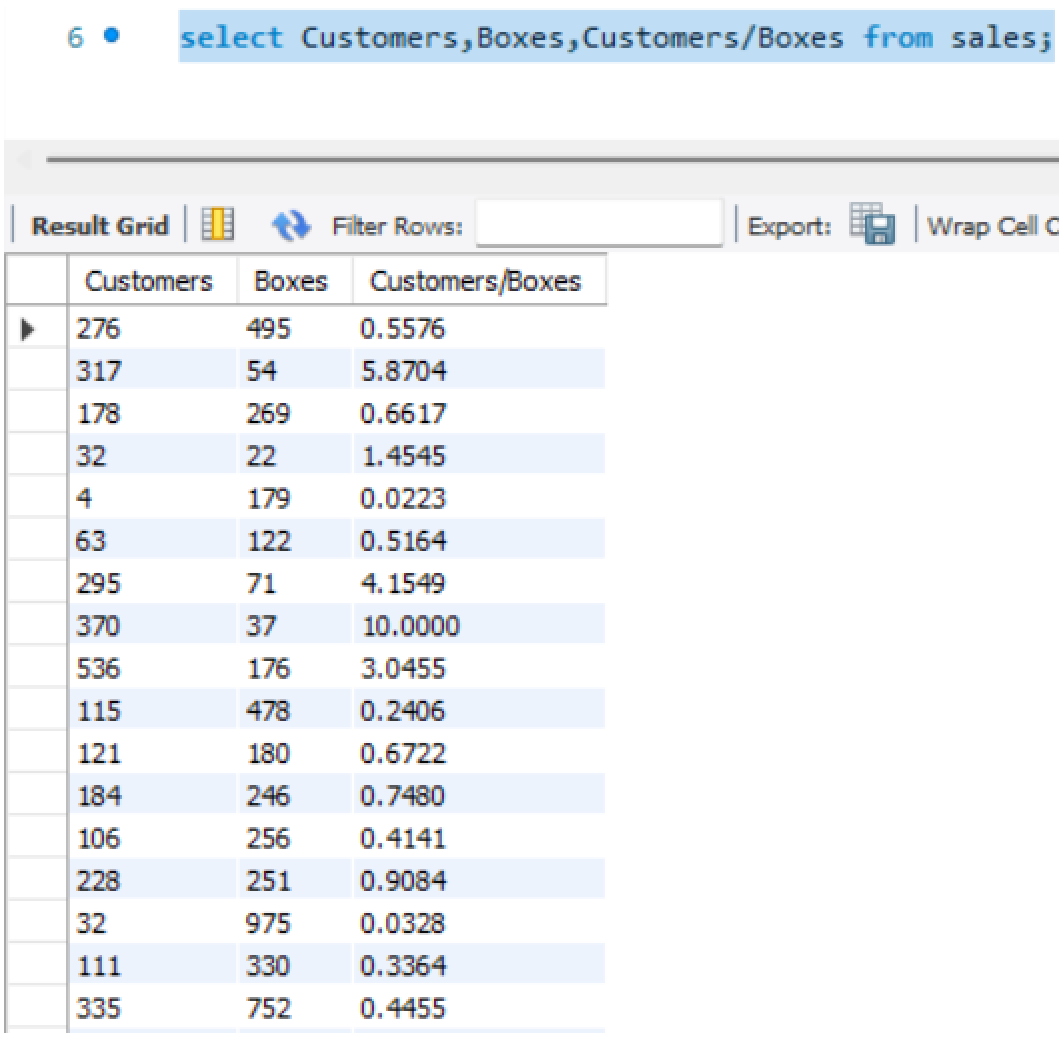
```sql
select SPID,Amount from sales;
```

SPID	Amount
SP01	8414
SP02	532
SP12	5376
SP01	259
SP19	5530
SP17	2184

Arithmetic operation

```sql
%%Syntax%%
    select colname,colname1(operator)colname2 from
tablename;
```

```sql
%%Example%%
    select Customers,Boxes,Customers/Boxes from sales;
```



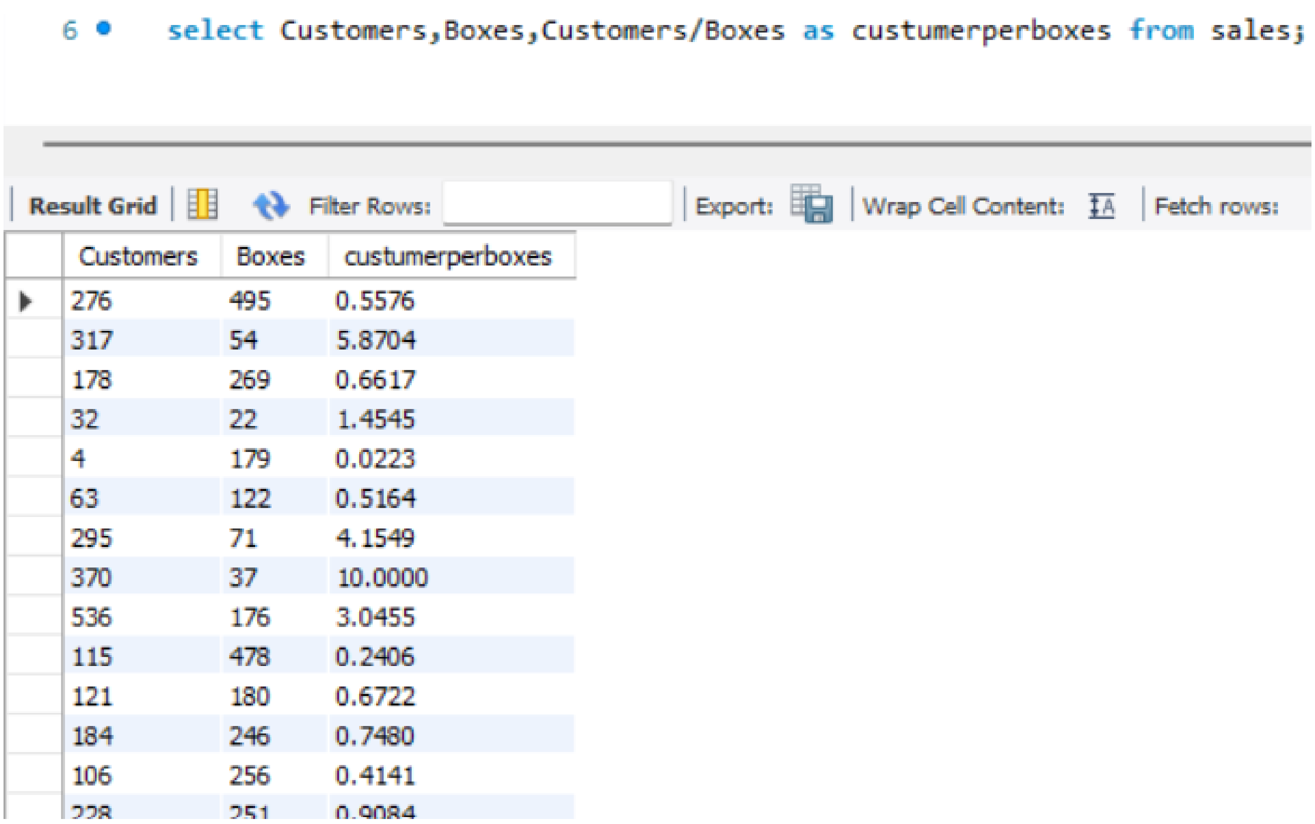
```
select Customers,Boxes,Customers/Boxes from sales;
```

Customers	Boxes	Customers/Boxes
276	495	0.5576
317	54	5.8704
178	269	0.6617
32	22	1.4545
4	179	0.0223
63	122	0.5164
295	71	4.1549
370	37	10.0000
536	176	3.0455
115	478	0.2406
121	180	0.6722
184	246	0.7480
106	256	0.4141
228	251	0.9084
32	975	0.0328
111	330	0.3364
335	752	0.4455

- adding names to operation

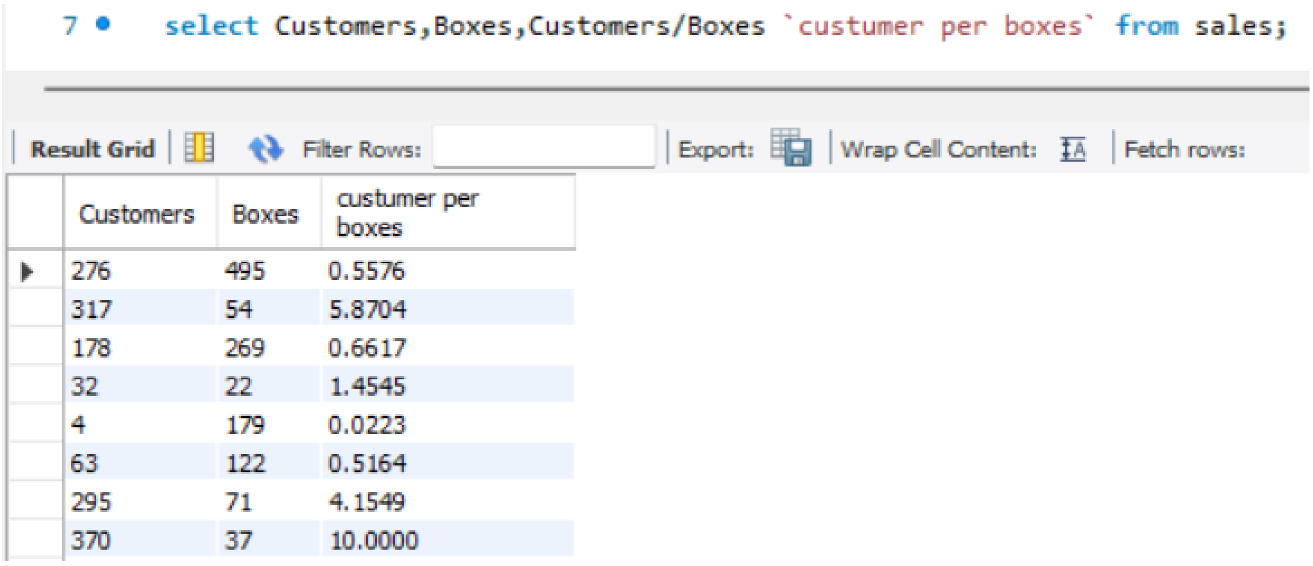
```sql
    select Customers,Boxes,Customers/Boxes as
custumerperboxes from sales;
```

```
    select Customers,Boxes,Customers/Boxes `custumer per
boxes` from sales;
```



```
select Customers,Boxes,Customers/Boxes as custumerperboxes from sales;
```

Customers	Boxes	custumerperboxes
276	495	0.5576
317	54	5.8704
178	269	0.6617
32	22	1.4545
4	179	0.0223
63	122	0.5164
295	71	4.1549
370	37	10.0000
536	176	3.0455
115	478	0.2406
121	180	0.6722
184	246	0.7480
106	256	0.4141
228	251	0.9084



```
select Customers,Boxes,Customers/Boxes `custumer per boxes` from sales;
```

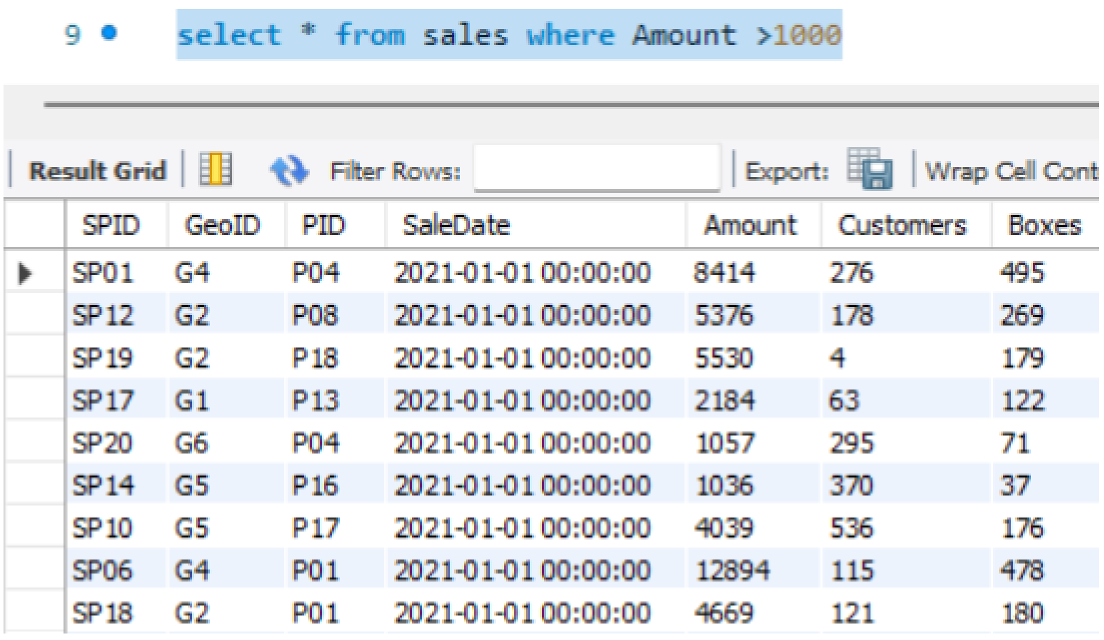
Customers	Boxes	custumer per boxes
276	495	0.5576
317	54	5.8704
178	269	0.6617
32	22	1.4545
4	179	0.0223
63	122	0.5164
295	71	4.1549
370	37	10.0000

Condition

The `WHERE` clause is used to filter records.

```SQL
%%Syntax%%
    select colname from table_name where condition;
```

```sql
%%Example%%
    select * from sales where Amount >1000;
```



```
select * from sales where Amount >1000
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP01	G4	P04	2021-01-01 00:00:00	8414	276	495
SP12	G2	P08	2021-01-01 00:00:00	5376	178	269
SP19	G2	P18	2021-01-01 00:00:00	5530	4	179
SP17	G1	P13	2021-01-01 00:00:00	2184	63	122
SP20	G6	P04	2021-01-01 00:00:00	1057	295	71
SP14	G5	P16	2021-01-01 00:00:00	1036	370	37
SP10	G5	P17	2021-01-01 00:00:00	4039	536	176
SP06	G4	P01	2021-01-01 00:00:00	12894	115	478
SP18	G2	P01	2021-01-01 00:00:00	4669	121	180

```sql
%%in SQL date format is yyyy-mm-dd%%
    select * from sales where (amount>10000 and SaleDate
>=
'2021-02-01');
    select saleDate,Amount from sales where (amount>10000
and month(saleDate)=02);
    select saleDate,Amount from sales where (amount>10000
and year(saleDate)=2021);
```

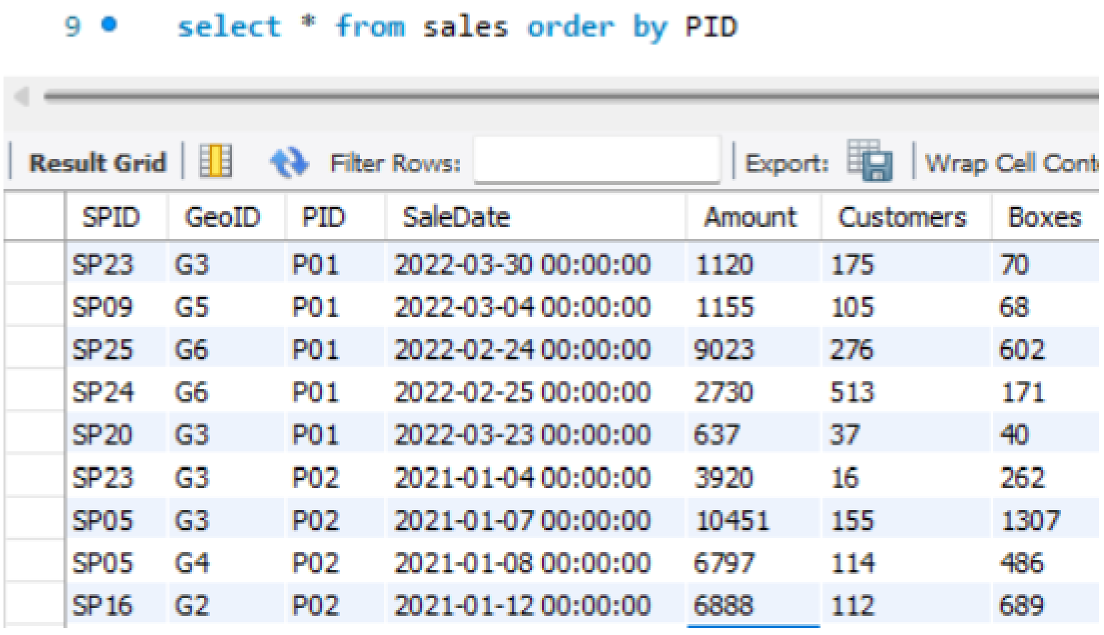
Order By

The `ORDER BY` keyword is used to sort the result-set in ascending or descending order. By default ascending order

```sql
%%Syntax%%
    %% for ascending order %%
        select colname from tablename order by colname;
    %% for descending order %%
        select colname from tablename order by colname
desc;
```

SQL

```
%%Example%%
    %%for ascending order%%
        select * from sales order by PID;
```



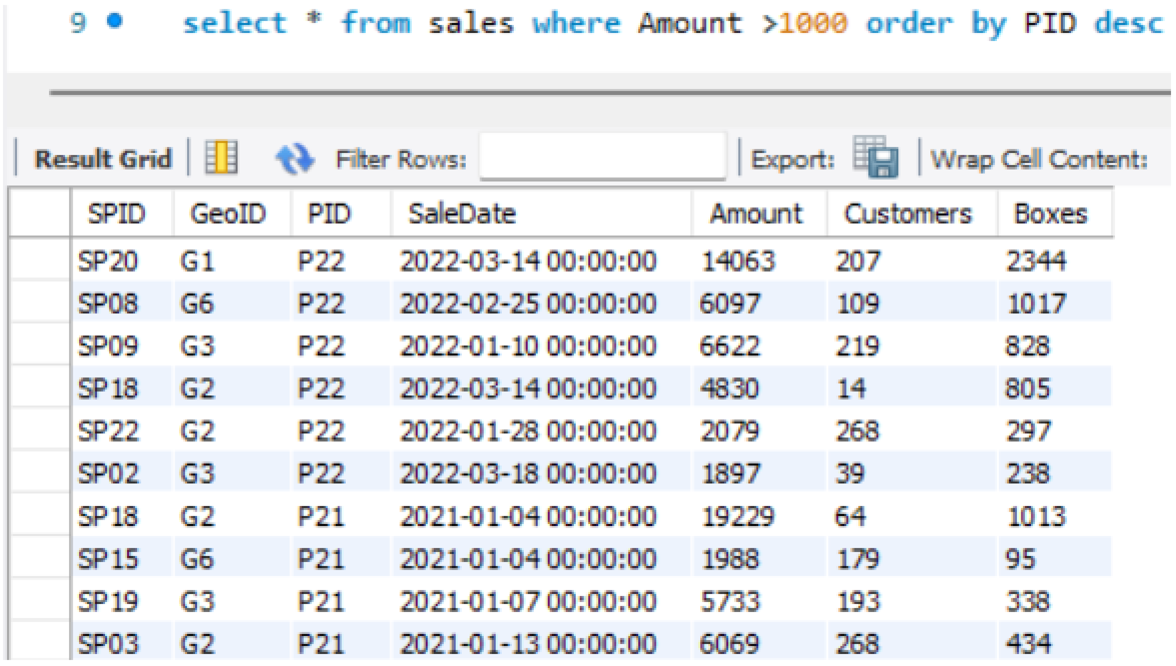
```
select * from sales order by PID
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP23	G3	P01	2022-03-30 00:00:00	1120	175	70
SP09	G5	P01	2022-03-04 00:00:00	1155	105	68
SP25	G6	P01	2022-02-24 00:00:00	9023	276	602
SP24	G6	P01	2022-02-25 00:00:00	2730	513	171
SP20	G3	P01	2022-03-23 00:00:00	637	37	40
SP23	G3	P02	2021-01-04 00:00:00	3920	16	262
SP05	G3	P02	2021-01-07 00:00:00	10451	155	1307
SP05	G4	P02	2021-01-08 00:00:00	6797	114	486
SP16	G2	P02	2021-01-12 00:00:00	6888	112	689

SQL

```
    %%for descending order%%
        select * from sales where Amount >1000 order by
PID desc;
```

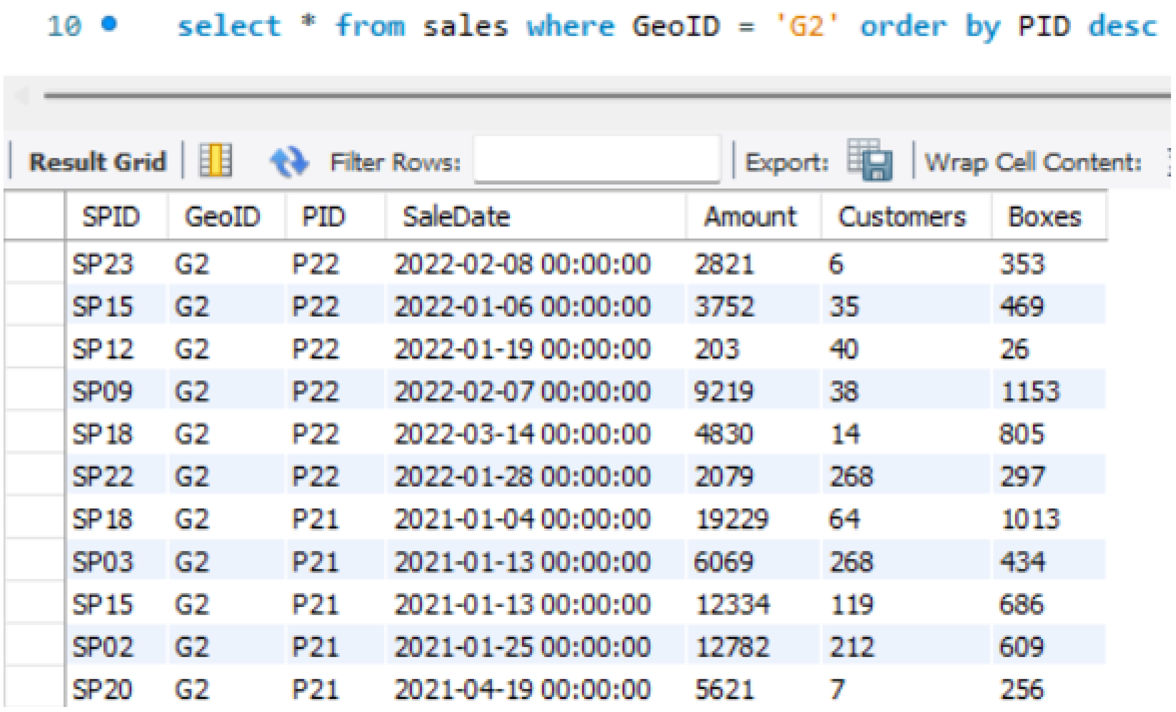


```
select * from sales where Amount >1000 order by PID desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP20	G1	P22	2022-03-14 00:00:00	14063	207	2344
SP08	G6	P22	2022-02-25 00:00:00	6097	109	1017
SP09	G3	P22	2022-01-10 00:00:00	6622	219	828
SP18	G2	P22	2022-03-14 00:00:00	4830	14	805
SP22	G2	P22	2022-01-28 00:00:00	2079	268	297
SP02	G3	P22	2022-03-18 00:00:00	1897	39	238
SP18	G2	P21	2021-01-04 00:00:00	19229	64	1013
SP15	G6	P21	2021-01-04 00:00:00	1988	179	95
SP19	G3	P21	2021-01-07 00:00:00	5733	193	338
SP03	G2	P21	2021-01-13 00:00:00	6069	268	434

```SQL
        select * from sales where GeoID = 'G2' order by
PID desc;
```



```
select * from sales where GeoID = 'G2' order by PID desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

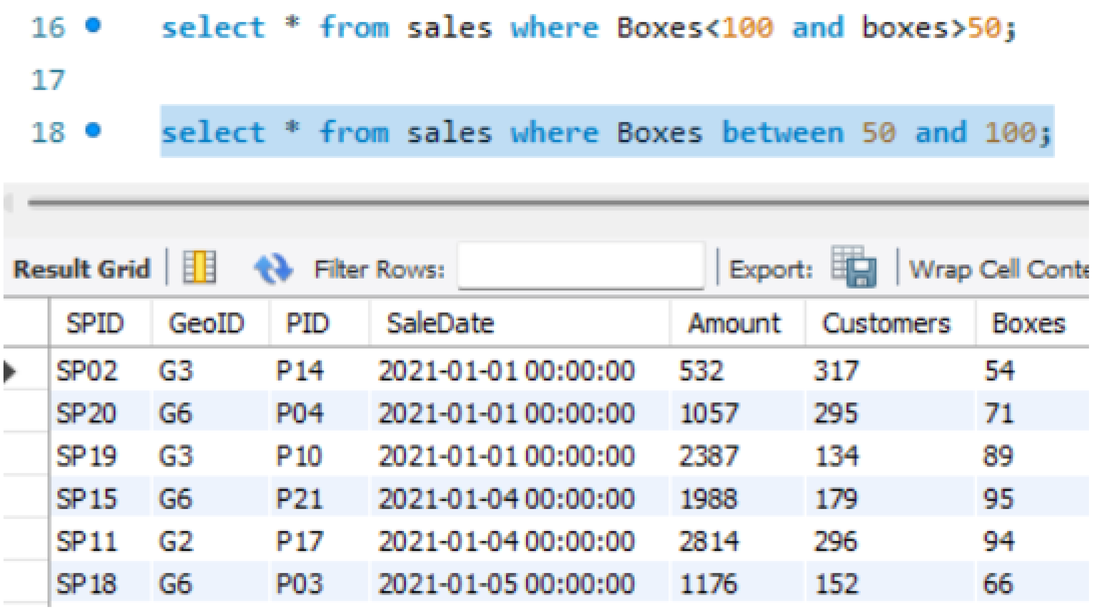
SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP23	G2	P22	2022-02-08 00:00:00	2821	6	353
SP15	G2	P22	2022-01-06 00:00:00	3752	35	469
SP12	G2	P22	2022-01-19 00:00:00	203	40	26
SP09	G2	P22	2022-02-07 00:00:00	9219	38	1153
SP18	G2	P22	2022-03-14 00:00:00	4830	14	805
SP22	G2	P22	2022-01-28 00:00:00	2079	268	297
SP18	G2	P21	2021-01-04 00:00:00	19229	64	1013
SP03	G2	P21	2021-01-13 00:00:00	6069	268	434
SP15	G2	P21	2021-01-13 00:00:00	12334	119	686
SP02	G2	P21	2021-01-25 00:00:00	12782	212	609
SP20	G2	P21	2021-04-19 00:00:00	5621	7	256

Between condition

```SQL
%%Syntax%%
    select col_name from table_name where (condition and
condition);
```

```
    select col_name from table_name where colname between
value1 and value2;
```

```SQL
%%Example%%
    select * from sales where Boxes<100 and boxes>50;
    select * from sales where Boxes between 50 and 100;
```



```
select * from sales where Boxes<100 and boxes>50;

select * from sales where Boxes between 50 and 100;
```

SPID	GeoID	PID	SaleDate	Amount	Customers	Boxes
SP02	G3	P14	2021-01-01 00:00:00	532	317	54
SP20	G6	P04	2021-01-01 00:00:00	1057	295	71
SP19	G3	P10	2021-01-01 00:00:00	2387	134	89
SP15	G6	P21	2021-01-04 00:00:00	1988	179	95
SP11	G2	P17	2021-01-04 00:00:00	2814	296	94
SP18	G6	P03	2021-01-05 00:00:00	1176	152	66

Date

- In sql `Weekday()` returns index for date monday=0 tuesday=1 sunday=6
- In sql `Year()` returns year part of the date

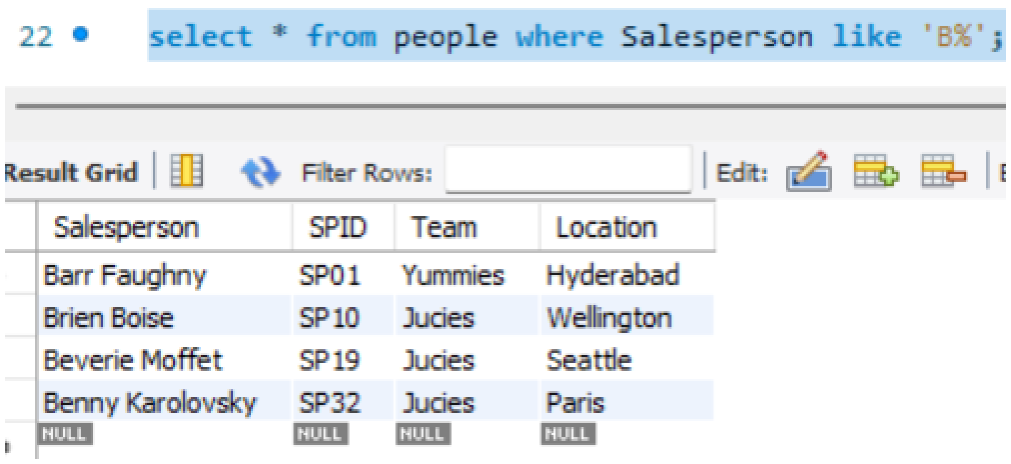
```SQL
%%Example%%
    select SPID, GeoID,PID, weekday(saleDate) as 'Day of
week' from sales;
```

Pattern matching

- The `LIKE` operator is used in a WHERE clause to search for a specified pattern in a column.

- The percent sign `(%)` represents zero, one, or multiple characters.
- The underscore sign `(_)` represents one, single character.

```SQL
%%Starting with B%%
    select * from people where Salesperson like 'B%';
%%Having B%%
    select * from people where Salesperson like '%B%';
```



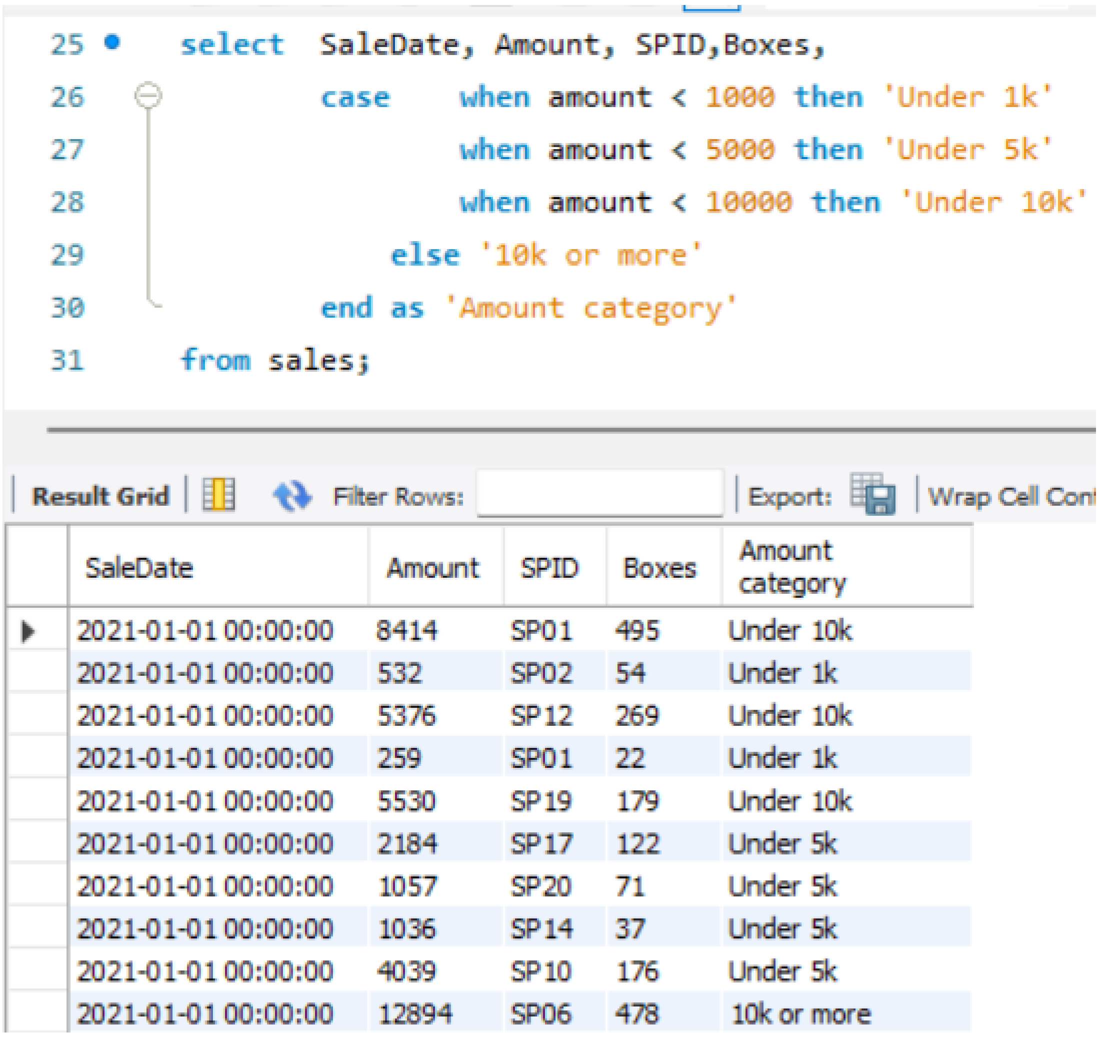
```
select * from people where Salesperson like 'B%';
```

Salesperson	SPID	Team	Location
Barr Faughny	SP01	Yummies	Hyderabad
Brien Boise	SP10	Jucies	Wellington
Beverie Moffet	SP19	Jucies	Seattle
Benny Karolovsky	SP32	Jucies	Paris
NULL	NULL	NULL	NULL

Case Operator and branching logic

- The `CASE` expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement).

```SQL
    select  SaleDate, Amount, SPID,Boxes,
            case    when amount < 1000 then 'Under 1k'
                    when amount < 5000 then 'Under 5k'
                    when amount < 10000 then 'Under 10k'
                else '10k or more'
            end as 'Amount category'
    from sales;
```



```
select   SaleDate, Amount, SPID,Boxes,
         case     when amount < 1000 then 'Under 1k'
                  when amount < 5000 then 'Under 5k'
                  when amount < 10000 then 'Under 10k'
              else '10k or more'
         end as 'Amount category'
from sales;
```

SaleDate	Amount	SPID	Boxes	Amount category
2021-01-01 00:00:00	8414	SP01	495	Under 10k
2021-01-01 00:00:00	532	SP02	54	Under 1k
2021-01-01 00:00:00	5376	SP12	269	Under 10k
2021-01-01 00:00:00	259	SP01	22	Under 1k
2021-01-01 00:00:00	5530	SP19	179	Under 10k
2021-01-01 00:00:00	2184	SP17	122	Under 5k
2021-01-01 00:00:00	1057	SP20	71	Under 5k
2021-01-01 00:00:00	1036	SP14	37	Under 5k
2021-01-01 00:00:00	4039	SP10	176	Under 5k
2021-01-01 00:00:00	12894	SP06	478	10k or more

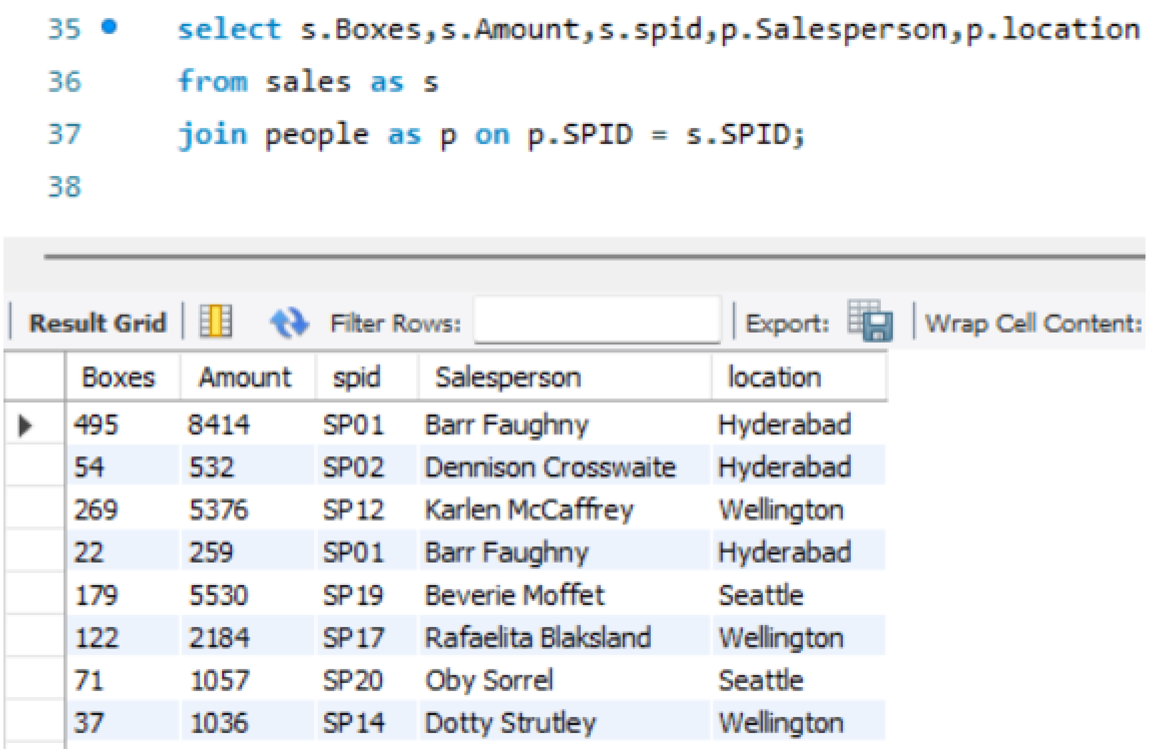
Join

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

```sql
%%Syntax%%
    SELECT columnsname
    FROM table1
    JOIN table2
    ON table1.column = table2.column;
```

```sql
%%Example%%
    select
s.Boxes,s.Amount,s.spid,p.Salesperson,p.location
```

```
    from sales as s
    join people as p on p.SPID = s.SPID;
```

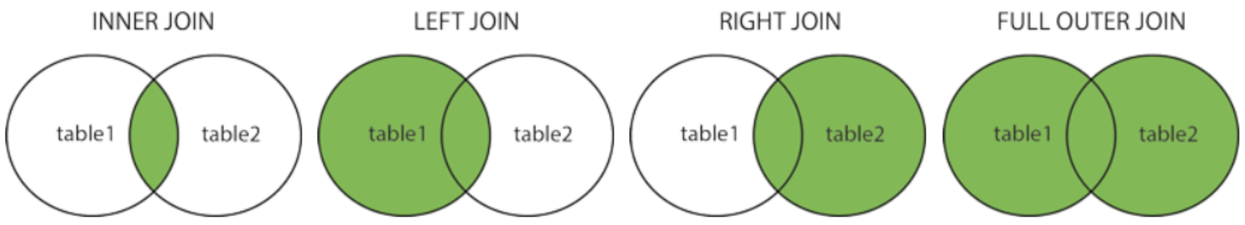


```
select s.Boxes,s.Amount,s.spid,p.Salesperson,p.location
from sales as s
join people as p on p.SPID = s.SPID;
```

Boxes	Amount	spid	Salesperson	location
495	8414	SP01	Barr Faughny	Hyderabad
54	532	SP02	Dennison Crosswaite	Hyderabad
269	5376	SP12	Karlen McCaffrey	Wellington
22	259	SP01	Barr Faughny	Hyderabad
179	5530	SP19	Beverie Moffet	Seattle
122	2184	SP17	Rafaelita Blaksland	Wellington
71	1057	SP20	Oby Sorrel	Seattle
37	1036	SP14	Dotty Strutley	Wellington

Types of the JOIN

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table

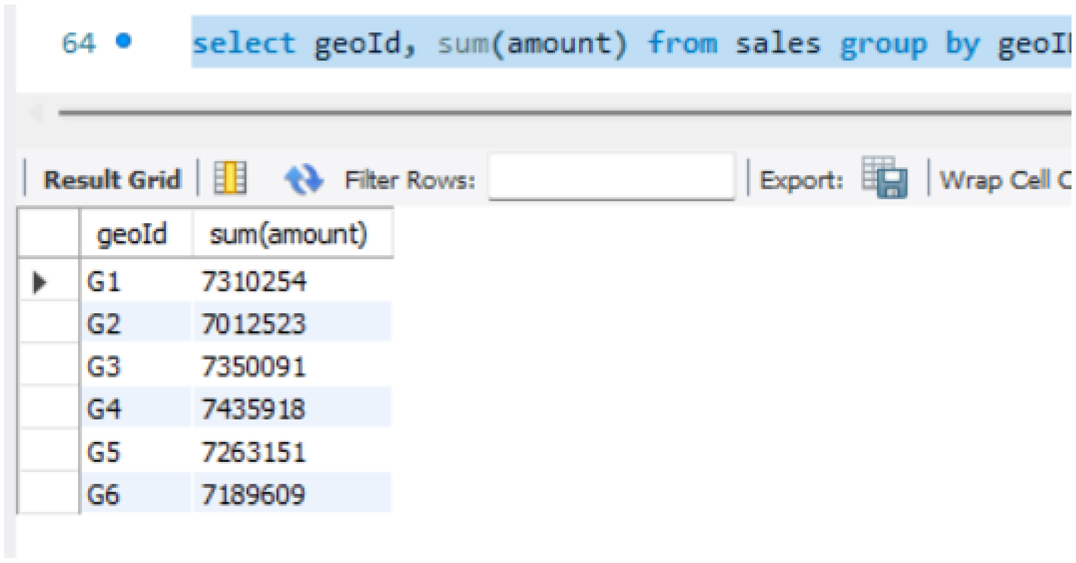


Group by

- The `GROUP BY` statement groups rows that have the same values into summary rows.

SQL

```sql
select geoId, sum(amount) from sales group by geoID
```



```
select geoId, sum(amount) from sales group by geoI
```

geoId	sum(amount)
G1	7310254
G2	7012523
G3	7350091
G4	7435918
G5	7263151
G6	7189609

Misc

SQL

```sql
select s.saleDate, s.amount, p.SalesPerson,
pr.product,g.Geo
from sales as s
join people as p on p.SPID = s.SPID
join products as pr on pr.pid = s.pid
join geo as g on g.GeoID = s.GeoID
where s.amount <1000
and pr.Product like 'White%'
and g.Geo in ('UK','Canada')
order by amount;
```