In [120]:
```
import pandas as pd
import numpy as np
```

Transacciones

A B C E
B E
C D E
A C D
A C E

In [121]:
```
transacciones = [['A','B','C','E'],['B','E'],['C','D','E'],['A','C','D'],['A','C'
transacciones
```

Out[121]:
```
[['A', 'B', 'C', 'E'],
 ['B', 'E'],
 ['C', 'D', 'E'],
 ['A', 'C', 'D'],
 ['A', 'C', 'E']]
```

itemset k = 1

In [122]:
```
a = []
b = []
c = []
d = []
e = []

i = 1
index = []
```

In [123]:
```
for tran in transacciones:
    a.append(tran.count('A'))
    b.append(tran.count('B'))
    c.append(tran.count('C'))
    d.append(tran.count('D'))
    e.append(tran.count('E'))

    index.append(str(i))
    i += 1
a
```

Out[123]:
```
[1, 0, 0, 1, 1]
```

In [124]:
```
result = {
    "ID": index,
    "A": a,
    "B": b,
    "C": c,
    "D": d,
    "E": e
}
```

Base de datos binaria

In [125]:
```
df = pd.DataFrame(result)
df
```

Out[125]:

	ID	A	B	C	D	E
0	1	1	1	1	0	1
1	2	0	1	0	0	1
2	3	0	0	1	1	1
3	4	1	0	1	1	0
4	5	1	0	1	0	1

Suma de ocurrencias de itemset y soporte

In [126]:
```
ocurra = sum(a)
ocurrb = sum(b)
ocurrc = sum(c)
ocurrd = sum(d)
ocurre = sum(e)
```

In [127]:
```
sopa = ocurra / 5
sopb = ocurrb / 5
sopc = ocurrc / 5
sopd = ocurrd / 5
sope = ocurre / 5
```

In [128]:
```
dict = {
    "item" : ["A", "B", "C", "D", "E"],
    "Ocurrencia" : [ocurra, ocurrb, ocurrc, ocurrd, ocurre],
    "Soporte" : [sopa, sopb, sopc, sopd, sope]
}

df = pd.DataFrame(dict)
df
```

Out[128]:

	item	Ocurrencia	Soporte
0	A	3	0.6
1	B	2	0.4
2	C	4	0.8
3	D	2	0.4
4	E	4	0.8

itemset k = 2

In [129]:
```
ab = []
ac = []
ad = []
ae = []
bc = []
bd = []
be = []
cd = []
ce = []
de = []

i = 1
index = []
```

In [130]:
```
for tran in transacciones:
    ab.append(1 if np.in1d(['A', 'B'], tran).all() else 0)
    ac.append(1 if np.in1d(['A', 'C'], tran).all() else 0)
    ad.append(1 if np.in1d(['A', 'D'], tran).all() else 0)
    ae.append(1 if np.in1d(['A', 'E'], tran).all() else 0)
    bc.append(1 if np.in1d(['B', 'C'], tran).all() else 0)
    bd.append(1 if np.in1d(['B', 'D'], tran).all() else 0)
    be.append(1 if np.in1d(['B', 'E'], tran).all() else 0)
    cd.append(1 if np.in1d(['C', 'D'], tran).all() else 0)
    ce.append(1 if np.in1d(['C', 'E'], tran).all() else 0)
    de.append(1 if np.in1d(['D', 'E'], tran).all() else 0)

    index.append(str(i))
    i += 1
ab
```

Out[130]:
```
[1, 0, 0, 0, 0]
```

In [131]:
```
resultado = {
    "ID": index,
    "AB": ab,
    "AC": ac,
    "AD": ad,
    "AE": ae,
    "BC": bc,
    "BD": bd,
    "BE": be,
    "CD": cd,
    "CE": ce,
    "DE": de
}
```

Base de datos binaria

In [132]:
```
df = pd.DataFrame(resultado)
df
```

Out[132]:

	ID	AB	AC	AD	AE	BC	BD	BE	CD	CE	DE
0	1	1	1	0	1	1	0	1	0	1	0
1	2	0	0	0	0	0	0	1	0	0	0
2	3	0	0	0	0	0	0	0	1	1	1
3	4	0	1	1	0	0	0	0	1	0	0
4	5	0	1	0	1	0	0	0	0	1	0

Suma de ocurrencias de itemset y soporte

In [133]:
```
ocurrab = sum(ab)
ocurrac = sum(ac)
ocurrad = sum(ad)
ocurrae = sum(ae)
ocurrbc = sum(bc)
ocurrbd = sum(bd)
ocurrbe = sum(be)
ocurrcd = sum(cd)
ocurrce = sum(ce)
ocurrde = sum(de)
```

In [134]:
```
sopab = ocurrab / 5
sopac = ocurrac / 5
sopad = ocurrad / 5
sopae = ocurrae / 5
sopbc = ocurrbc / 5
sopbd = ocurrbd / 5
sopbe = ocurrbe / 5
sopcd = ocurrcd / 5
sopce = ocurrce / 5
sopde = ocurrde / 5
```

In [135]:
```
dict = {
    "item" : ["{A,B}", "{A,C}", "{A,D}", "{A,E}", "{B,C}", "{B,D}", "{B,E}", "{C,
    "Ocurrencia" : [ocurrab, ocurrac, ocurrad, ocurrae, ocurrbc, ocurrbd, ocurrbe
    "Soporte" : [sopab, sopac, sopad, sopae, sopbc, sopbd, sopbe, sopcd, sopce, s
}
```

In [136]:
```
df = pd.DataFrame(dict)
df
```

Out[136]:

	item	Ocurrencia	Soporte
0	{A,B}	1	0.2
1	{A,C}	3	0.6
2	{A,D}	1	0.2
3	{A,E}	2	0.4
4	{B,C}	1	0.2
5	{B,D}	0	0.0
6	{B,E}	2	0.4
7	{C,D}	2	0.4
8	{C,E}	3	0.6
9	{D,E}	1	0.2

Se **descartan** los itemset que no superen el soporte mínimo (<=0.5)

In [137]:
```
filt = (df["Soporte"] >= 0.5)
df[filt]
```

Out[137]:

	item	Ocurrencia	Soporte
1	{A,C}	3	0.6
8	{C,E}	3	0.6

itemset k = 3

In [138]:
```
ace = []

i = 1
index = []
```

In [139]:
```
for tran in transacciones:
    ace.append(1 if np.in1d(['A', 'C', 'E'], tran).all() else 0)

    index.append(str(i))
    i += 1
ace
```

Out[139]:
```
[1, 0, 0, 0, 1]
```

In [140]:
```
resultado = {
    "ID": index,
    "ACE": ace,
}
```

Base de datos binaria

In [141]:
```
df = pd.DataFrame(resultado)
df
```

Out[141]:

	ID	ACE
0	1	1
1	2	0
2	3	0
3	4	0
4	5	1

Suma de ocurrencias de itemset y soporte

In [142]:
```
ocurrace = sum(ace)
```

In [143]:
```
sopace = ocurrace / 5
```

In [144]:
```
dict = {
    "item" : ["{A,C,E}"],
    "Ocurrencia" : [ocurrace],
    "Soporte" : [sopace]
}
```

In [145]:
```
df = pd.DataFrame(dict)
df
```

Out[145]:

	item	Ocurrencia	Soporte
0	{A,C,E}	2	0.4

Se **descartan** los itemset que no superen el soporte mínimo (<=0.5)

In [146]:
```
filt = (df["Soporte"] >= 0.5)
df[filt]
```

Out[146]:

item	Ocurrencia	Soporte

La tabla regresa vacía, es decir, **ningún soporte** supera el umbral mínimo

Se detiene el algoritmo

Confianza >= 0.7

In [147]:
```
reglaac1 = sopac / sopa
reglaac2 = sopac / sopc
reglace1 = sopce / sopc
reglace2 = sopce / sope
```

In [148]:
```
dict = {
    "item" : ["{A,C} 1", "{A,C} 2", "{C,E} 1", "{C,E} 2"],
    "Regla" : ["{A} => {C} 1", "{A} => {C} 2", "{C} => {E} 1", "{C} => {E} 2"],
    "Soporte" : [reglaac1,reglaac2,reglace1,reglace2]
}
```

In [149]:
```
df = pd.DataFrame(dict)
df
```

Out[149]:

	item	Regla	Soporte
0	{A,C} 1	{A} => {C} 1	1.00
1	{A,C} 2	{A} => {C} 2	0.75
2	{C,E} 1	{C} => {E} 1	0.75
3	{C,E} 2	{C} => {E} 2	0.75

En conclusión, el nivel más alto al que se puede llegar en este caso de transacciones es **K=2** con un soporte(umbral) mínimo de **0.5**, donde todas las reglas de asociación son altamente confiables, pues todas son mayor o igual al **75% de confianza**.