In [3]:
```
import pandas as pd
import sqlite3
data = pd.read_csv('New_Wireless_Pipe.txt', sep="|")
cnn = sqlite3.connect('juypter_sql_tutorial.db')
# data.to_sql('df', cnn)
%load_ext sql
%sql sqlite:///juypter_sql_tutorial.db
```

In [4]:
```
# Q1: check if the data has duplicate records
# spark.sql("""
#
# """).show()
# +------+
# |is_dup|
# +------+
# | false|
# +------+
```

In [5]:
```
%%sql

SELECT (case when max(count) =1 then 'false' ELSE 'true' end ) AS is_dup
FROM(
SELECT acctno, count(acctno) AS count
FROM df
GROUP BY acctno) AS t
```

```
 * sqlite:///juypter_sql_tutorial.db
Done.
```

Out[5]:

is_dup
false

In [ ]:
```
# Q2: plase list 2nd place sales account no. for each dealertype; then how to c
# spark.sql("""
#
#
# """).show()
# +----------+-------------+-----+----+
# |dealertype|       acctno|sales|rank|
# +----------+-------------+-----+----+
# |        A2|1879712055400| 1200|   2|
# |        B1|2111710057053| 1200|   2|
# |        C1|2344491947376| 1199|   2|
# |        A1|1595694049314| 1200|   2|
# +----------+-------------+-----+----+
```

In [ ]:
```
%%sql
SELECT * FROM(
SELECT dealertype, acctno, sales, row_number() OVER(PARTITION BY dealertype ORI
FROM df
ORDER BY sales DESC)
WHERE rank = 2

# rank = 相同的同一排名
# row_num = 相同值 按顺序排a
```

In [ ]:
```
# +----------+-------------+-----+----+
# |dealertype|       acctno|sales|rank|
# +----------+-------------+-----+----+
# |        A2|1329951214050| 1200|   1|
# |        A2|1879712055400| 1200|   2|
# |        A2|1415944669550| 1199|   3|
# |        B1|1538431934945| 1200|   1|
# |        B1|2111710057053| 1200|   2|
# |        B1|2220633276528| 1199|   3|
# |        C1|1556976795264| 1199|   1|
# |        C1|2344491947376| 1199|   2|
# |        C1|2389693376124| 1199|   3|
# |        A1|1538055998888| 1200|   1|
# |        A1|1595694049314| 1200|   2|
# |        A1|1897085618697| 1200|   3|
# +----------+-------------+-----+----+
```

In [ ]:
```
%%sql
SELECT * FROM(
SELECT dealertype, acctno, sales, row_number() OVER(PARTITION BY dealertype ORI
FROM df
ORDER BY sales DESC)
WHERE rank <= 3
ORDER BY dealertype
```

In [ ]:
```
# Q3, display a list of rateplan, where the most popular rate plan on the top,
# spark.sql("""
#
# """).show()
# +--------+---------+-----------+
# |rateplan|use_count|total_sales|
# +--------+---------+-----------+
# |       1|    68194|   11306589|
# |       2|    20187|    3384503|
# |       3|    13874|    2282611|
# +--------+---------+-----------+
```

In [ ]:
```
%%sql

SELECT rateplan, count(rateplan) AS use_count, sum(sales) AS total_sales
FROM df
GROUP BY rateplan
ORDER BY count(rateplan) DESC
```

In [ ]:
```
# Q4 get the dealertype for rateplan's user more than 10,000, sort descending c
# spark.sql("""
#
# """).show()
# +----------+--------+----------+
# |dealertype|rateplan|user_count|
# +----------+--------+----------+
# |        A1|       1|     35875|
# |        B1|       1|     14239|
# |        A1|       2|     11837|
# |        C1|       1|     10105|
# +----------+--------+----------+

# Let's duplicate the data and do some dedup exercise
```

In [ ]:
```
%%sql

SELECT dealertype, rateplan, count(rateplan) AS user_count
FROM df
GROUP BY dealertype, rateplan
HAVING count(rateplan) >10000
ORDER BY count(rateplan) DESC
```

In [ ]:
```
# Q5, get the number of rows,the num of acct and the num of unique accounts usi
# #use sql to display
# df_stat = spark.sql("""
#
# """)
# df_stat.show()
# +------+------+----------+
# |   Obs| Accts|Uniq_Accts|
# +------+------+----------+
# |102265|102265|    102255|
# +------+------+----------+
```

In [ ]:
```
%%sql

SELECT COUNT(*) AS Obs, COUNT(acctno) AS Accts, COUNT(DISTINCT acctno) AS Uniq_
FROM df
```

In [ ]:
```
# Q6 HOW TO CREATE A DATASET WITHOUT DUPLICATE RECORD?
# think about how to do that via pandas and sql

# df_uniq = spark.sql("""
#
# """)
# print('the record count for the dataframe is :',df_uniq.count())
# the record count for the dataframe is : 102255
```

In [ ]:
```
df = """
    SELECT COUNT(DISTINCT acctno) AS count
    FROM df
    PRINT
"""

df_uniq = pd.read_sql_query(df, cnn)
print('the record count for the dataframe is :',df_uniq['count'][0])
```

In [ ]:
```
# Q7 HOW TO CREATE A DATASET WITH ALL THE DUPLICATED RECORD?
#
# df_dup = spark.sql("""
#
# """)
# df_dup.show()
# +------------+----------+----------+-----------+----------+--------+--------
# |       acctno|     actdt|   deactdt|deactreason|goodcredit|rateplan|dealerty
# +------------+----------+----------+-----------+----------+--------+--------
# |1176913194483|06/20/1999|      null|       null|         0|       1|
# |1176913194483|06/20/1999|      null|       null|         0|       1|
# |1176951913656|07/01/2000|      null|       null|         0|       1|
# |1176951913656|07/01/2000|      null|       null|         0|       1|
# |1176991056273|08/31/1999|09/18/2000|       MOVE|         1|       1|
```

```
# |1176991056273|08/31/1999|09/18/2000|       MOVE|          1|       1|
# |1176991866552|05/24/2000|      null|       null|          1|       1|
# |1176991866552|05/24/2000|      null|       null|          1|       1|
# |1176992889500|11/28/2000|      null|       null|          1|       1|
# |1176992889500|11/28/2000|      null|       null|          1|       1|
# |1177010940613|12/09/1999|      null|       null|          1|       2|
# |1177010940613|12/09/1999|      null|       null|          1|       2|
# |1176954000288|05/30/2000|      null|       null|          1|       2|
# |1176954000288|05/30/2000|      null|       null|          1|       2|
# |1176914599423|10/04/1999|10/15/1999|       NEED|          1|       1|
# |1176914599423|10/04/1999|10/15/1999|       NEED|          1|       1|
# |1176969186303|12/13/2000|      null|       null|          1|       1|
# |1176969186303|12/13/2000|      null|       null|          1|       1|
# |1177000067271|12/23/1999|      null|       null|          0|       1|
# |1177000067271|12/23/1999|      null|       null|          0|       1|
# +-------------+----------+----------+-----------+-----------+--------+--------
```

In [ ]:

```
%%sql

SELECT t.acctno, t.actdt, t.deactdt, t.deactreason, t.goodcredit, t.rateplan, t
FROM df1 AS s
INNER JOIN (
SELECT *, count(*) AS count
FROM df1
GROUP BY acctno
HAVING COUNT(*) >1) AS t
ON s.acctno = t.acctno
```