SQL for Analytics
Data Tables

IOE 373 Lecture 08

Topics

- Preparing a Data Analysis Table
- Date functions
- Recency, Longevity, Mean Time Between Purchases/Transactions

Data Preparation

- Just as manufacturing and refining are about transformation of raw materials into finished products…
- With analytics we need to extract, clean, transform our data to prepare for analysis…

What the Data Should Look Like

- All data in a single table (rows/columns)
- Each row corresponds to an entity (e.g. customer)
- Columns with single/unique value should be ignored (where the same value is observed for all rows)
- Target column identified for predictive analysis (e.g. output variable, what we want to predict)

What the Data Should Look Like

- Customer (Entity) <u>Signature</u>
  - Continuous "snapshot" of customer behavior

Each row represents the customer and whatever might be useful to include in a model

This column is an id field where the value is different in every column. It gets ignored for data mining purposes.

This column is from the customer information file.

This column is the target, what we want to predict.

2610000101	010377	14		A	19.1		14 Spring . . .	TRUE
2610000102	103188	7		A	19.1		NULL	TRUE
2610000105	041598	1		B	21.2		71 W. 19 St.	FALSE
2610000171	040296	1		S	38.3		3562 Oak. . .	FALSE
2610000182	051990	22		C	56.1		9672 W. 142	FALSE
2610000183	111192	45		C	56.1		NULL	TRUE
2620000107	080891	6		A	19.1		P.O. Box 11	FALSE
2620000108	120398	3		D	10.0		560 Robson	TRUE
2620000220	022797	2		S	38.3		222 E. 11th	FALSE
2620000221	021797	3		A	19.1		10122 SW 9	FALSE
2620000230	060899	1		S	38.3		NULL	TRUE
2620000231	062099	10		S	38.3		RR 1729	TRUE
2620000300	032894	7		B	21.2		1920 S. 14th	FALSE

These rows have invalid customer ids, so they are ignored.

This column is summarized from transaction data.

This column is a text field with unique values. It gets ignored (although it may be used for some derived variables).

What the Data Should Look Like

- Columns have important Model Roles in Data Modeling/Analysis:
  - Input columns – input into the model
  - Target column(s) – used only for predictive models – the values are created by the algorithm
  - Ignored columns – not used in a particular data mining analysis

Case Study: Automotive Sales Data Analysis Table

- Need to generate a data set for analysis with various factors (inputs or predictors) and one response variable (target) per **Household id**.
  - Predictors:
    - NumCustomers per Household
    - ZipHHMedianIncome
    - **NumOrders**
    - **Recency(days)**
    - **Longevity(days)**
    - **Mean Time between purchases (days)**
    - NumOrders/Percentage (campaign 2173)
  - Target:
    - Indicator Variable (has the household ordered under campaign 2173, yes or no? (1 or 0)

Cleansing/Checking Duplicates

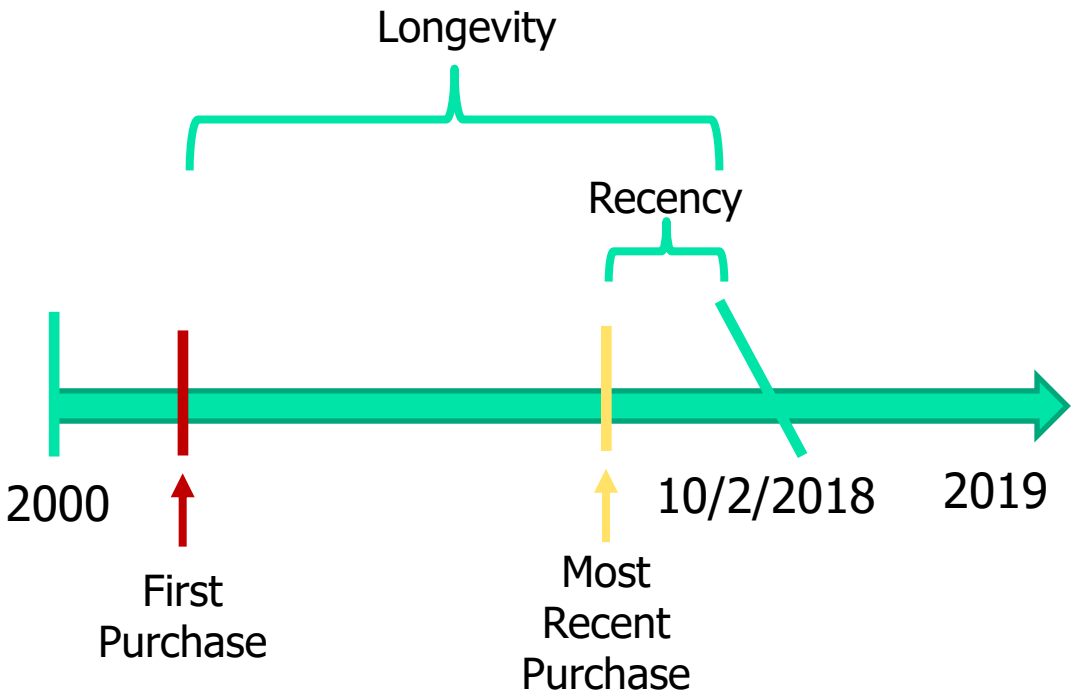
customerid	householdid	gender	firstname
5	18792705		J.
6	19842675	M	JERRY
7	20425541	F	GRETCHEN
8	24916819		
9	18467513	F	JOAN
10	19840635	F	HANNE
11	20680937		B.
12	18861747		ROMAN
13	19321406		P.
14	18701167	F	KATHLEEN
15	19793375		J.
16	18370825	M	FRANK
17	19334762	F	MADELINE
18	21298187		HENRY
19	19327075	F	ROBERTA
20	18158507	M	JOSEPH
21	19159162	M	BENEDICT
22	19613746	M	ARMAND
23	20634581	F	LINDA
24	19467302		E.
25	19467302		E.
26	19467302		E.

Removing duplicates

```
SELECT MIN([Copy Of Customer].customerid)
AS customerid, [Copy Of
Customer].householdid, [Copy Of
Customer].gender, [Copy Of
Customer].firstname, COUNT(*) as NumCopies
INTO Customer
FROM [Copy Of Customer]
GROUP BY  [Copy Of Customer].householdid,
[Copy Of Customer].gender, [Copy Of
Customer].firstname
```

customerid	householdid	gender	firstname	NumCopies
1	20516395	F	EILEEN	1
2	20346368	F	ENID	1
3	19038361	F	ANNA	1
4	18572491		L.	1
5	18792705		J.	1
6	19842675	M	JERRY	1
7	20425541	F	GRETCHEN	1
9	18467513	F	JOAN	1
10	19840635	F	HANNE	1
11	20680937		B.	1
12	18861747		ROMAN	1
13	19321406		P.	1
14	18701167	F	KATHLEEN	1
15	19793375		J.	1
16	18370825	M	FRANK	1
17	19334762	F	MADELINE	1
18	21298187		HENRY	1
19	19327075	F	ROBERTA	1
20	18158507	M	JOSEPH	1
21	19159162	M	BENEDICT	1
22	19613746	M	ARMAND	1
23	20634581	F	LINDA	1
24	19467302		E.	3

Time related variables



DATEDIFF Function

- Calculates the difference between 2 dates.
- Syntax:
  - DateDiff (interval, date1, date2)

DateDiff ("yyyy", #15/10/1998#, #22/11/2003#)
Result: 5

DateDiff ("m", #15/10/2003#, #22/11/2003#)
Result: 1

DateDiff ("d", #15/10/2003#, #22/11/2003#)
Result: 38

Interval	Explanation
yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week
h	Hour
n	Minute
s	Second

DateValue Function

- Converts a string to a date.
- Sintax:
  - DateValue (string_date)

Recency

Select householdid, MAX(orderdate) as MaxDate, DATEDIFF("d", MaxDate, #10/2/2018#) as Recency_days

FROM orders INNER JOIN customer ON orders.customerid=customer.customerid

GROUP BY householdid

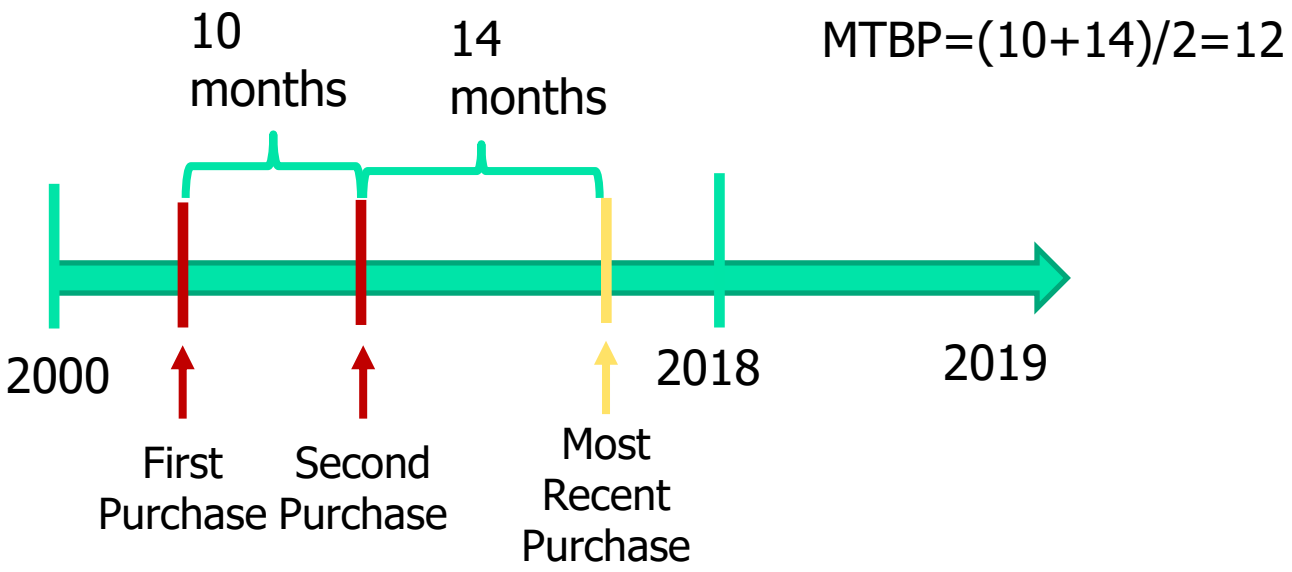
Recency

```
SELECT householdid, MAX(orderdate) as maxdate,
DATEDIFF("d", maxdate, #10/2/2018#) as
recency_days
    FROM orders INNER JOIN customer ON
orders.customerid = customer.customerid
    WHERE orderdate < datevalue("10/02/2018")
    GROUP BY householdid
ORDER BY householdid;
```

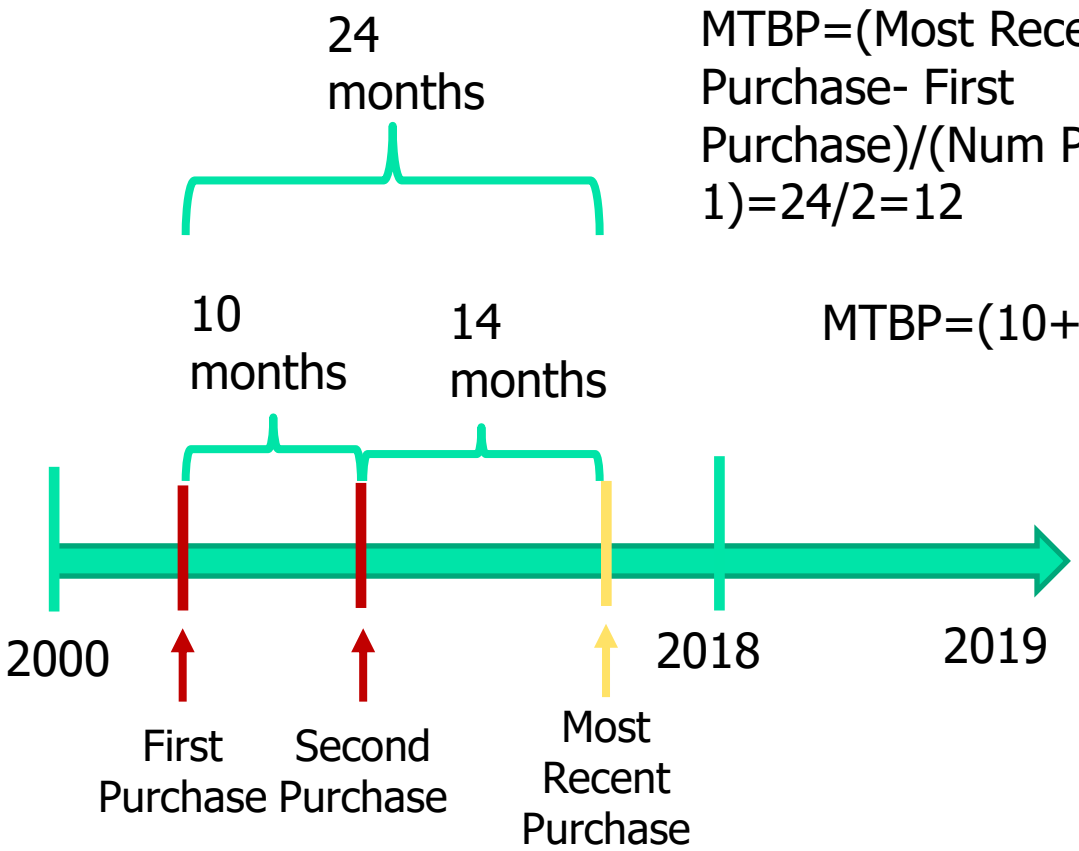
Longevity

```
SELECT householdid, MIN(orderdate) as mindate,
DATEDIFF("d", mindate, datevalue("10/02/2018")) as longevity_days
    FROM orders INNER JOIN customer ON orders.customerid = customer.customerid
    WHERE orderdate < datevalue("10/02/2018")
    GROUP BY householdid
ORDER BY householdid;
```

Mean Time Between Purchases



Mean Time Between Purchases



MTBP=(Most Recent Purchase- First Purchase)/(Num Purchases-1)=24/2=12

MTBP=(10+14)/2=12

Mean Time Between Purchases

```
SELECT householdid, MIN(orderdate) as mindate,
            MAX(orderdate) as maxdate, COUNT(*) as
numorders, round(DATEDIFF("d", mindate,
maxdate) / (numorders - 1), 0) as
MeanTimeBP_Days
    FROM orders as o INNER JOIN customer as c ON
o.customerid = c.customerid
    WHERE orderdate < datevalue("10/02/2018")
    GROUP BY householdid
    HAVING COUNT(*) > 1
ORDER BY householdid
```

Full Table (Recency, Longevity, MTBP)

```
SELECT householdid, count(*) as NumCustomersHH, MIN(orderdate) as mindate,
        MAX(orderdate) as maxdate, COUNT(*) as numorders, DATEDIFF("d", maxdate, datevalue("10/02/2018")) as recency_days, DATEDIFF("d", mindate, datevalue("10/02/2018")) as longevity_days, round(DATEDIFF("d", mindate, maxdate) / (numorders - 1), 0) as MeanTime_Days
    FROM orders as o INNER JOIN customer as c ON o.customerid = c.customerid
    WHERE orderdate < datevalue("10/02/2018")
    GROUP BY householdid
    HAVING COUNT(*) > 1
ORDER BY householdid
```

Full Table (ZipCensus Info)

SELECT householdid, count(c.customerid) as NumCustomersHH, Max(z.hhmedincome) AS ZipHHMedIncome, **MIN(orderdate) as mindate, MAX(orderdate) as maxdate, count(*) as numorders, DATEDIFF("d", maxdate, datevalue("10/02/2018")) as recency_days, DATEDIFF("d", mindate, datevalue("10/02/2018")) as longevity_days, round(DATEDIFF("d", mindate, maxdate) / (numorders - 1), 0) as MeanTime_Days**

FROM (orders AS o INNER JOIN customer AS c ON o.customerid = c.customerid) INNER JOIN Zipcensus As z ON o.zipcode=z.zipcode

WHERE orderdate < datevalue("10/02/2018")

GROUP BY householdid

HAVING COUNT(*) > 1

ORDER BY householdid

Full Table (Create Final Table with target variable)

SELECT householdid, count(c.customerid) as NumCustomersHH, Max(z.hhmedincome) AS ZipHHMedIncome, MIN(orderdate) as mindate, MAX(orderdate) as maxdate, COUNT(*) as numorders, DATEDIFF("d", maxdate, datevalue("09/30/2014")) as recency_days, DATEDIFF("d", mindate, datevalue("09/30/2014")) as longevity_days, round(DATEDIFF("d", mindate, maxdate) / (numorders - 1), 0) as MeanTime_Days, SUM(IIF(o.campaignid=2173, 1, 0)) as NumCampaign2173, SUM(IIF(o.campaignid=2173, 1, 0))/Count(*)*100 AS PercentCampaign2173, MAX(IIF(o.campaignid=2173, 1, 0)) as Campaign2173IND INTO FinalTable

FROM (orders AS o INNER JOIN customer AS c ON o.customerid = c.customerid) INNER JOIN Zipcensus As z ON o.zipcode=z.zipcode

WHERE orderdate < datevalue("10/02/2018")

GROUP BY householdid

HAVING COUNT(*) > 1 And Count(*)<=10

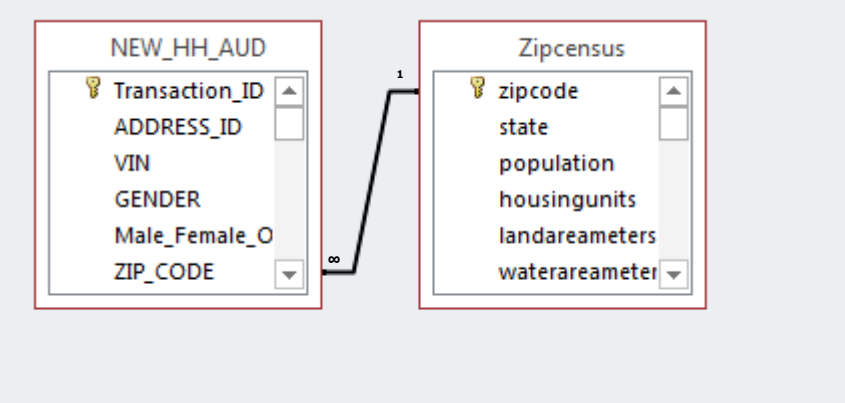
ORDER BY c.householdid;

Final Table

householdid	Num	ZipHHMedIn	mindate	maxdate	numc	recency_day	longevity_da	MeanTime_[	Num(	Percent	Camp
18122420	2	117487	11/26/2012	6/17/2014	2	105	673	568	0	0	0
18127110	2	73553	2/10/2011	11/14/2011	2	1051	1328	277	0	0	0
18132047	2	111492	11/5/2009	12/4/2011	2	1031	1790	759	1	50	1
18140431	2	93785	8/17/2010	4/17/2014	2	166	1505	1339	0	0	0
18141105	2	72165	5/29/2013	2/16/2014	2	226	489	263	0	0	0
18149246	2	74903	12/28/2011	12/12/2012	2	657	1007	350	0	0	0
18154604	2	44518	9/5/2011	6/11/2012	2	841	1121	280	0	0	0
18156898	2	57132	1/18/2011	2/7/2011	2	1331	1351	20	0	0	0
18166768	2	128524	3/12/2011	4/30/2014	2	153	1298	1145	0	0	0
18176069	2	60043	9/22/2010	12/12/2011	2	1023	1469	446	1	50	1
18180758	2	71295	12/2/2011	3/28/2014	2	186	1033	847	1	50	1
18181625	2	91527	11/19/2012	4/15/2013	2	533	680	147	0	0	0
18182422	2	68368	12/15/2012	12/3/2013	2	301	654	353	0	0	0
18185812	2	61472	12/6/2010	8/20/2014	2	41	1394	1353	0	0	0
18194342	2	60672	11/17/2009	12/14/2010	2	1386	1778	392	1	50	1
18194384	2	58379	6/26/2011	2/7/2012	2	966	1192	226	0	0	0
18198908	2	62932	3/29/2012	11/30/2012	2	669	915	246	0	0	0
18204124	2	105971	6/14/2010	12/14/2011	2	1021	1569	548	0	0	0
18211103	2	67285	11/17/2010	9/21/2011	2	1105	1413	308	2	100	1

Case Study: Car Sales A4 Model (ExampleDB.acccdb)

- Need to generate a data set for analysis with various factors (inputs or predictors) and one response variable (target) per ADDRESS_ID.
  - Predictors:
    - Total Audi Vehicles
    - Percent Audi Vehicles
    - Total VW Vehicles
    - Percent VW Vehicles
    - Total New
    - Percent New
    - Recency(months)
    - Longevity(months)
    - Mean Time between purchases (months)
    - Is the Address in an area with MedianIncome>$30,000 (1 or 0)
  - Target:
    - A4 Indicator Variable (has the household purchased an A4, yes or no? (1 or 0)



Remove Duplicates

```
SELECT FORMAT(MIN(NEW_HH_AUD_OLD.ADDRESS_ID),"#") AS ADDRESS_ID, NEW_HH_AUD_OLD.VIN,
NEW_HH_AUD_OLD.GENDER, NEW_HH_AUD_OLD.Male_Female_Other, NEW_HH_AUD_OLD.ZIP_CODE,
NEW_HH_AUD_OLD.VEHICLE_SEGMENT, NEW_HH_AUD_OLD.SALE_DATE, NEW_HH_AUD_OLD.SALE_YEAR,
NEW_HH_AUD_OLD.DISPOSAL_DATE, NEW_HH_AUD_OLD.SALE_MDL_YEAR,
NEW_HH_AUD_OLD.SALE_BRAND, NEW_HH_AUD_OLD.BODY_NAME, NEW_HH_AUD_OLD.BODY_CATG,
NEW_HH_AUD_OLD.KIND_OF_SALE, NEW_HH_AUD_OLD.DEALER_CODE,
NEW_HH_AUD_OLD.DEALER_REGION, NEW_HH_AUD_OLD.DEALER_CNTY,
NEW_HH_AUD_OLD.BASIC_WRNTY_END_DATE, NEW_HH_AUD_OLD.ADDI_WRNTY_END_DATE,
NEW_HH_AUD_OLD.LEASE_OR_PURCHASE, NEW_HH_AUD_OLD.CLAIM_COUNT,
NEW_HH_AUD_OLD.RECALL_COUNT, NEW_HH_AUD_OLD.[Still Owned?],
NEW_HH_AUD_OLD.MonthsSincePurchase, NEW_HH_AUD_OLD.NewUsed,
NEW_HH_AUD_OLD.ExtendedWarranty, COUNT(*) AS NUM_COPIES INTO NEW_HH_AUD
FROM NEW_HH_AUD_old
GROUP BY NEW_HH_AUD_old.VIN, NEW_HH_AUD_OLD.GENDER, NEW_HH_AUD_OLD.Male_Female_Other,
NEW_HH_AUD_OLD.ZIP_CODE, NEW_HH_AUD_OLD.VEHICLE_SEGMENT, NEW_HH_AUD_OLD.SALE_DATE,
NEW_HH_AUD_OLD.SALE_YEAR, NEW_HH_AUD_OLD.DISPOSAL_DATE, NEW_HH_AUD_OLD.SALE_MDL_YEAR,
NEW_HH_AUD_OLD.SALE_BRAND, NEW_HH_AUD_OLD.BODY_NAME, NEW_HH_AUD_OLD.BODY_CATG,
NEW_HH_AUD_OLD.KIND_OF_SALE, NEW_HH_AUD_OLD.DEALER_CODE,
NEW_HH_AUD_OLD.DEALER_REGION, NEW_HH_AUD_OLD.DEALER_CNTY,
NEW_HH_AUD_OLD.BASIC_WRNTY_END_DATE, NEW_HH_AUD_OLD.ADDI_WRNTY_END_DATE,
NEW_HH_AUD_OLD.LEASE_OR_PURCHASE, NEW_HH_AUD_OLD.CLAIM_COUNT,
NEW_HH_AUD_OLD.RECALL_COUNT, NEW_HH_AUD_OLD.[Still Owned?],
NEW_HH_AUD_OLD.MonthsSincePurchase, NEW_HH_AUD_OLD.NewUsed,
NEW_HH_AUD_OLD.ExtendedWarranty
ORDER BY COUNT(*) DESC;
```

Analysis Table

```
SELECT NEW_HH_AUD.ADDRESS_ID, count(*) AS TOT_VEH,
SUM(IIF(NEW_HH_AUD.SALE_BRAND="AUD",1,0)) AS TOT_AUD,
SUM(IIF(NEW_HH_AUD.SALE_BRAND="AUD",1,0))/count(*) AS PERCENT_AUD,
SUM(IIF(NEW_HH_AUD.SALE_BRAND="VLK",1,0)) AS TOT_VLK,
SUM(IIF(NEW_HH_AUD.SALE_BRAND="VLK",1,0))/count(*) AS PERCENT_VLK,
SUM(IIF(NEW_HH_AUD.KIND_OF_SALE="New", 1,0)) AS Tot_NEW,
SUM(IIF(NEW_HH_AUD.KIND_OF_SALE="New", 1,0))/count(*) AS Percent_New,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A4",1,0)) AS Tot_A4,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A4",1,0))/count(*) AS Percent_A4,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A6",1,0)) AS Tot_A6,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A6",1,0))/count(*) AS Percent_A6,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A3",1,0)) AS Tot_A3,
SUM(IIF(NEW_HH_AUD.BODY_NAME="A3",1,0))/count(*) AS Percent_A3,
datediff("m",max(SALE_DATE),now()) AS RecencyMonths, datediff("m",min(SALE_DATE),
now()) AS LongevityMonths, round(DATEDIFF("m", min(SALE_DATE), max(SALE_DATE)) /
(TOT_VEH - 1), 0) AS MTBP, Max(IIF(Zipcensus.hhmedincome>30000,1,0)) AS
ZipMedIncome_30, MAX(IIF(BODY_NAME="A4",1,0)) AS Target_A4_Ind
FROM NEW_HH_AUD INNER JOIN Zipcensus ON NEW_HH_AUD.ZIP_CODE=Zipcensus.zipcode
GROUP BY NEW_HH_AUD.ADDRESS_ID
HAVING COUNT(*)>1 AND COUNT(*) <=10;
```

Final Table

ADDRESS_ID	TOT_VEH	TOT_AUD	PERCENT_AUD	TOT_VLK	PERCENT_VLK	Tot_NEW	Percent_New	Tot_A4	Percent_A4	Tot_A6	Percent_A6	Tot_A3	Percent_A3	RecencyMon	LongevityMo	MTBP	ZipMedIncor	Target_A4_Ir
100127498986965000	2	1	0.5	1	0.5	1	0.5	1	0.5	0	0	0	0	92	132	40	1	1
100127512834526000	2	2	1	0	0	0	0	1	0.5	0	0	0	0	113	150	37	1	1
100127568209423000	7	1	0.142857142857143	6	0.857142857142857	7	1	1	0.142857142857143	0	0	0	0	102	131	5	1	1
100127568480657000	2	2	1	0	0	2	1	1	0.5	0	0	0	0	96	152	56	1	1
100127568496640000	2	2	1	0	0	1	0.5	2	1	0	0	0	0	122	148	26	1	1
100127568519173000	2	2	1	0	0	2	1	2	1	0	0	0	0	97	124	27	1	1
100127569770182000	2	2	1	0	0	2	1	0	0	0	0	0	0	95	145	50	1	0
100127									1	0	0	0	0	137	137	0	1	1
100127									1	0	0	0	0	115	150	35	1	1
100127									0.5	1	0.5	0	0	99	135	36	1	1
100127									0	0	0	0	0	134	149	15	1	0
									0	0	0	0	0	99	135	36	1	0
									0.5	0	0	0	0	95	130	35	1	1