

## Access Aggregation Example

Reduce 108,791 loan application records into 430 aggregate records based on interest rate.

### Access Design View

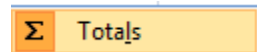
The screenshot shows the Access Design View for a query named "LoanStatsForExcel". The design grid is as follows:

Field:	Interest Rate	ID	CREDIT Grade	CREDIT Grade	Application Date	Application Date
Table:	LoanStatsForExcel	LoanStatsForExcel	LoanStatsForExcel	LoanStatsForExcel	LoanStatsForExcel	LoanStatsForExcel
Total:	Group By	Count	Min	Max	Min	Max
Sort:	Ascending					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:						
or:						

The field list on the left includes: ID (primary key), Loan ID, Amount Requested, Amount Funded By Investor, Interest Rate, Loan Length, Application Date, Application Expiration Date, Issued Date, CREDIT Grade, Loan Title, Loan Purpose, Loan Description, and Monthly Payment.

Sort by all GroupBy variables.

To specify aggregate functions in Access design view, right click in "design area" and select:



Aggregate functions: **Sum, Avg, Min, Max, Count, StDev, Var, First, Last, Expression, Where**

### SQL View

```
SELECT LoanStatsForExcel.[Interest Rate], Count(LoanStatsForExcel.ID) AS CountOfID, Min(LoanStatsForExcel.[CREDIT Grade]) AS [MinOfCREDIT Grade], Max(LoanStatsForExcel.[CREDIT Grade]) AS [MaxOfCREDIT Grade],
Min(LoanStatsForExcel.[Application Date]) AS [MinOfApplication Date], Max(LoanStatsForExcel.[Application Date]) AS [MaxOfApplication Date]
FROM LoanStatsForExcel
GROUP BY LoanStatsForExcel.[Interest Rate]
ORDER BY LoanStatsForExcel.[Interest Rate];
```

### Access query result:

Interest Rate	CountOfID	MinOfCREDI	MaxOfCREDI	MinOfAppli	MaxOfAppli
	3				
5.42	573	A1	A1	10/18/2010	9/8/2011
5.79	410	A2	A2	10/18/2010	4/25/2011
5.99	347	A2	A2	4/26/2011	9/22/2011

...

## Aggregation in R using split, do.call, rbind, lapply and an anonymous function

```
# Read data.frame
> d <- read.csv("LoanStats.csv", skip=1, as.is=TRUE)
> dim(d)
[1] 108789    42

# Create "splits" by Interest.Rate value
> splits <- split(d, d$Interest.Rate)
> length(splits)
[1] 429

# Look at first split
> x <- splits[[1]]
> dim(x)
[1] 573    42

# Look at how to aggregate
> x$Interest.Rate[1]
[1] 5.42
> nrow(x)
[1] 573
> min(x$CREDIT.Grade)
[1] "A1"
> max(x$CREDIT.Grade)
[1] "A1"
> min(x$Application.Date)
[1] "2010-10-18"
> max(x$Application.Date)
[1] "2011-09-08"

# Use "anonymous" function and rbind with do.call to create aggregate data.frame
> Data.By.Split <- do.call(rbind, lapply(splits,
+                               function(x)
+                                 {
+                                   data.frame(
+                                     Interest.Rate=x$Interest.Rate[1],
+                                     N=nrow(x),
+                                     minCreditGrade=min(x$CREDIT.Grade),
+                                     maxCreditGrade=max(x$CREDIT.Grade),
+                                     minApplDate=min(x$Application.Date),
+                                     maxApplDate=max(x$Application.Date),
+                                     stringsAsFactors=FALSE)
+                                 }
+                               ))
> row.names(Data.By.Split) <- 1:nrow(Data.By.Split)
> head(Data.By.Split)
  Interest.Rate    N minCreditGrade maxCreditGrade minApplDate maxApplDate
1         5.42  573             A1             A1  2010-10-18  2011-09-08
2         5.79  410             A2             A2  2010-10-18  2011-04-25
3         5.99  347             A2             A2  2011-04-26  2011-09-22
4         6.00   30             A4             G1  2008-01-19  2012-07-20
5         6.03 2367             A1             A1  2011-09-09  2013-02-04
6         6.17  252             A3             A3  2010-10-18  2011-01-19
```