Aggregate functions in SQL

Count ()

Sum ()

Max()

Min ()

Avg()

FirstName	LastName	Age
John	Smith	18
Jeremy	Smith	23
Mark	Long	64
Bob	James	37
Adam	Marcos	41

Count(*)	Sum(Age)	Max(Age)	Min(Age)	Avg(Age)
	18	18	18	18
	23	23	23	23
	64	64	64	64
	37	37	37	37
	41	41	41	41
5	**183**	**64**	**18**	**36**

Aggregate (examples)

Select Count(*), Sum(Age), Max (Age)

From **Bank.Client**;

Select Count(*), Sum(Age), Max (Age)

From **Bank.Client**

Where Type = ‘private’;

Aggregate (examples)

```
Select ClientId, Max(Balance), Min(Balance)
From Bank.Accounts
Where  Type = 'CREDIT'
Group by ClientId;
```

```
Select ClientId, Avg(Balance), Sum(Balance)
From Bank.Accounts
Group by ClientId
Having Avg(Balance) > 10000;
```

Join

SQL JOIN is used to combine rows from two or more tables, based on a common field between them.

JOIN (inner join)

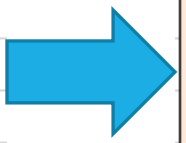
The INNER JOIN keyword selects records that have matching values in both tables.

```
SELECT *
FROM HomePro.Customers
    join HomePro.Schedules
    on Customers.CustomerId = Schedules.CustomerId
```

Customers							Schedules				
CustomerId	FirstName	LastName	Email	Phone	ZipCode	NewsLetter	Id	CustomerId	Description	DateNeeded	JobType
1	John	Smith	John@gmail.com	703-543-3302	22201	1	1	1	Kitchen remodel needed	2013-10-10	Remodeling
2	Jeremy	Smith	Jeremy@gmail.com	723-543-3302	22203	0	2	2	Decorationg help for dinig room	2013-10-15	Decorating
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	22031	1	3	3	Kitchen remodel needed	2015-11-29	Remodeling
3	Mark	Long	MarkLong@Yahoo.com	722-366-5588	22031	1	4	3	Garade rebuild	2016-12-31	Rebuild
4	Bob	James	bob@microsoft.com	703-366-9632	22221	0					
5	Adam	Marcos	adam@Marcos.com	703-566-0000	22001	1					

How it works

CustomerId	FirstName	LastName
1	John	Smith
2	Jeremy	Smith
3	Mark	Long
4	Bob	James
5	Adam	Marcos



Id	CustomerId	Description	DateNeeded
1	1	Kitchen remodel needed	2013-10-10
2	2	Decorationg help for dinig room	2013-10-15
3	3	Kitchen remodel needed	2015-11-29
4	3	Garade rebuild	2016-12-31

CustomerId	FirstName	LastName	Id	CustomerId	Description	DateNeeded
1	John	Smith	1	1	Kitchen remodel needed	2013-10-10
2	Jeremy	Smith	2	2	Decorationg help for dinig room	2013-10-15
3	Mark	Long	3	3	Kitchen remodel needed	2015-11-29
3	Mark	Long	4	3	Garade rebuild	2016-12-31
~~4~~	~~Bob~~	~~James~~				
~~5~~	~~Adam~~	~~Marcos~~				

Using table and column alias

```
SELECT
    a.CustomerId, a.FirstName as CustomerName, a.Lastname, b.Description as JobDescription, b.DateNeeded
FROM HomePro.Customers a
    join HomePro.Schedules b
    on a.CustomerId = b.CustomerId
```

HomePro.Customers				HomePro.Schedules	
CustomerId	CustomerName	LastName	Phone	JobDescription	DateNeeded
1	John	Smith	703-543-3302	Kitchen remodel needed	2013-10-10
2	Jeremy	Smith	723-543-3302	Decorationg help for dinig room	2013-10-15
3	Mark	Long	722-366-5588	Kitchen remodel needed	2015-11-29
3	Mark	Long	722-366-5588	Garade rebuild	2016-12-31
4	Bob	James	703-366-9632		
5	Adam	Marcos	703-566-0000		

Left join

returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

```
SELECT
c.CustomerId, c.FirstName, c.Lastname, c.Phone, s.Description, s.DateNeeded
FROM HomePro.Customers c
    left join HomePro.Schedules s
    on  c.CustomerId = s.CustomerId
```

HomePro.Customers				HomePro.Schedules	
CustomerId	FirstName	LastName	Phone	Description	DateNeeded
1	John	Smith	703-543-3302	Kitchen remodel needed	2013-10-10
2	Jeremy	Smith	723-543-3302	Decorationg help for dinig room	2013-10-15
3	Mark	Long	722-366-5588	Kitchen remodel needed	2015-11-29
3	Mark	Long	722-366-5588	Garade rebuild	2016-12-31
4	Bob	James	703-366-9632	NULL	NULL
5	Adam	Marcos	703-566-0000	NULL	NULL

Customers without Schedules

```
SELECT
 c.CustomerId, c.FirstName, c.Lastname, c.Phone, s.Description, s.DateNeeded
FROM HomePro.Customers c
    left join HomePro.Schedules s
    on c.CustomerId = s.CustomerId
WHERE s.CustomerId is null
```

Customers				Schedules	
CustomerId	FirstName	LastName	Phone	Description	DateNeeded
4	Bob	James	703-366-9632	NULL	NULL
5	Adam	Marcos	703-566-0000	NULL	NULL

Right join

Returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

```
SELECT
  s.Description, s.DateNeeded, c.CustomerId, c.FirstName, c.Lastname
FROM HomePro.Schedules s
   right join HomePro.Customers c
   on c.CustomerId = s.CustomerId
```

HomePro.Schedules		HomePro.Customers		
Description	DateNeeded	CustomerId	LastName	LastName
Kitchen remodel needed	2013-10-10	1	John	Smith
Decorationg help for dinig room	2013-10-15	2	Jeremy	Smith
Kitchen remodel needed	2015-11-29	3	Mark	Long
Garade rebuild	2016-12-31	3	Mark	Long
NULL	NULL	4	Bob	James
NULL	NULL	5	Adam	Marcos

Full outer join

Combines the result of both LEFT and RIGHT joins

```
SELECT
  c.CustomerId, c.FirstName, c.Lastname, c.Phone, s.id, s.CustomerId,
  s.Description, s.DateNeeded
FROM HomePro.Customers c
   full outer join HomePro.Schedules s
   on c.CustomerId = s.CustomerId
```

HomePro.Customers			HomePro.Schedules			
CustomerId	FirstName	LastName	Id	CustomerId	Description	DateNeeded
1	John	Smith	1	1	Kitchen remodel needed	2013-10-10
2	Jeremy	Smith	2	2	Decorationg help for dinig room	2013-10-15
3	Mark	Long	3	3	Kitchen remodel needed	2015-11-29
3	Mark	Long	4	3	Garade rebuild	2016-12-31
4	Bob	James	NULL	NULL	NULL	NULL
5	Adam	Marcos	NULL	NULL	NULL	NULL
NULL	NULL	NULL	5	6	Kitchen remodel needed	2013-10-10

Three tables join

```
SELECT
   C.CustomerId, C.FirstName, C.Lastname, S.Description, S.DateNeeded, Q.Estimation
FROM HomePro.Customers C
   join HomePro.Schedules S on C.CustomerId = S.CustomerId
   join HomePro.Quotes Q on C.CustomerId = Q.CustomerId
```

Three tables left join

```
SELECT
 C.CustomerId, C.FirstName, C.Lastname, S.Description, S.DateNeeded, Q.Estimation
FROM HomePro.Customers C
     left join HomePro.Schedules S on C.CustomerId = S.CustomerId
     left join HomePro.Quotes Q on C.CustomerId = Q.CustomerId
```