Understanding Joins in Ad Hoc Reports

What is a join?

When you use more than one data table in the Ad Hoc report writer, you need to join the tables together using a common value in both sets of data. You are able to connect the data from these tables in different ways, and the ways you can connect them constitute the various joins.

Why are joins important to my report?

Often times the data that you need for your report does not exist in a single table. Two or more tables need to be connected so that you can construct your report. There may not be a one to one correspondence between the tables that will be connected. For example, a single permit often has many fees – this is a one to many relationship, or a single permit may have no completed inspections – this is a one to none relationship. The data that you are able to pull into your report will depend on the type of join you select. If you join two tables of data in an incompatible way, you may get inaccurate results or no results at all.

What types of joins exist?

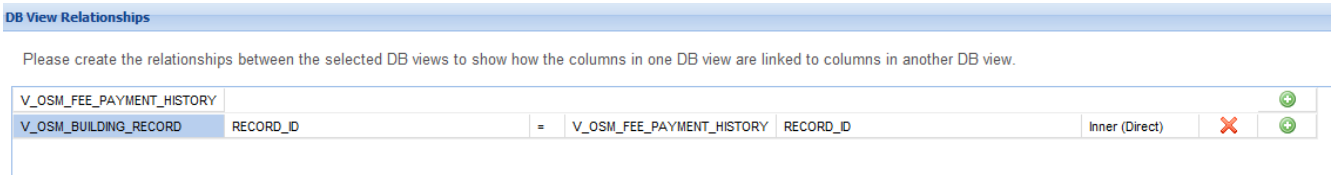
Accela has five types of joins available in Ad Hoc: the inner (direct) join, the cross join, the left (first exists) join, the right join, and the full join.

How do I know which join to use?

Being familiar with the different types of joins, as well as being familiar with your data, will allow you to choose to the best possible join type. The inner join or the left join will probably be what you use most often. Read about the different types of joins and contact our helpdesk if you have any questions.

How to Join Tables

First, select the two tables you want to use in your report. The join view, called DB View Relationships, will automatically pop up when a second table is checked. Here is a screen shot of what DB View Relationships will look like after you fill in the cells using the dropdown menus:

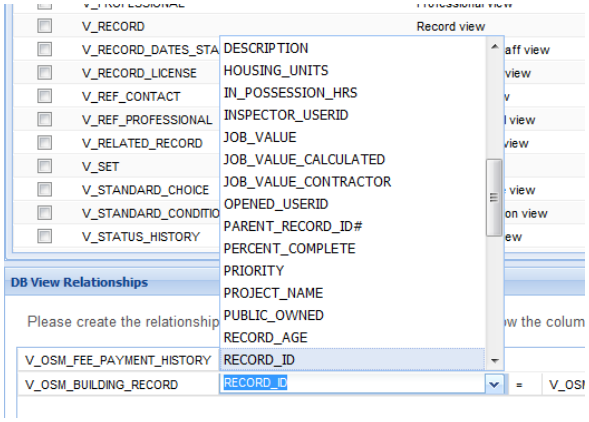


For this example, the V_OSM_FEE_PAYMENT_HISTORY table has automatically populated in the top left field, because it was the first table selected in the checkboxes. The field directly below it is the first cell, where you use a dropdown menu to select a data table to create the join. The second table that was checked, V_OSM_BUILDING_RECORD in the example, is the table you will want to select for that first cell.

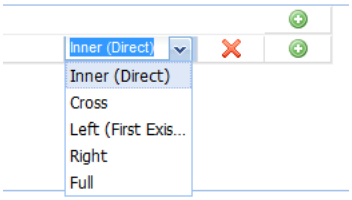
In the second cell, you will select a value from the second data source. The value you select here needs to be able to be mapped to the column you will select in the fifth cell. We will explain this more as we cover joins below. In the example, we have selected RECORD_ID.

The third cell is a read-only operator that contains the equal sign.

The fourth cell should automatically populate with the name of the first data table you selected, V_OSM_FEE_PAYMENT_HISTORY in our screen shot example. The fifth cell is for selecting a column from the first data source that you are mapping data to from the second data source. In our example, a column in both of the chosen data tables (V_OSM_FEE_PAYMENT_HISTORY and V_OSM_BUILDING_RECORD) is RECORD_ID. RECORD_ID is the column that contains the record numbers assigned to permits, planning applications, ect. Because that column is the same in both tables, the tables can joined through this column.



The dropdown menu in the second cell



The sixth cell is where you will identify how you want to join the tables. You can join the data in specific ways to either include or exclude certain data from your report.

Terminology

A **database** is a collection of information arranged into tables. A **table** is a set of data that has been grouped together based on relevance. A **field** is a single piece of data in a table.

Here is an example of a database that we will use as we further explain joins:
Bob's Sports Outlet – System Database

Customers Table

Account #	Name	Address	Phone
1	John Smith	563 Maple St.	503-373-7396
2	Kerry Jones	1212 Ivy Ln.	503-373-7396
3	Reggie Brown	1623 Elm St.	503-373-7396
4	Sam Spade	909 Poplar Rd.	503-373-7396

Purchases Table

Name	Product Name	Product Qty	Purchase Price
John Smith	Basketball	2	$24.00
Kerry Jones	Football	1	$23.00
Kerry Jones	Football Helmet	2	$69.00
Reggie Brown	Baseball	1	$20.00
Reggie Brown	Baseball Hat	5	$45.00
Guest	Football	2	$46.00

Vendors Table

Vendor Name	Vendor Product
Rawlins Sports	Football
Rawlins Sports	Baseball
Rawlins Sports	Softball
Rawlins Sports	Basketball
Riggins Supplies	Basketball Net
Riggins Supplies	Soccer Ball

Products Table

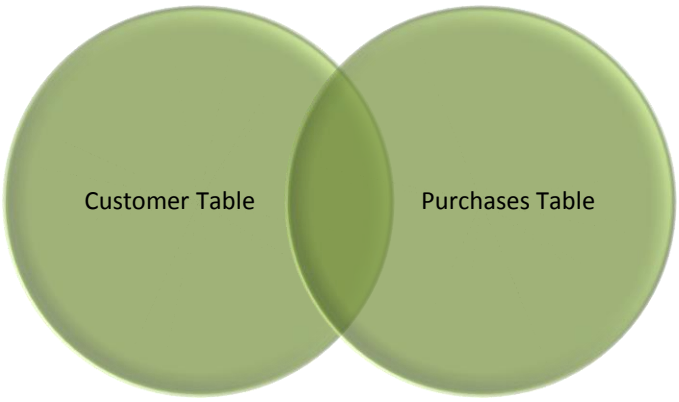
Product #	Description	Price
1	Basketball	$23.00
2	Football	$29.00
3	Baseball hat	$70.00
4	Baseball mitt	$73.00
5	Baseball helmet	$69.00
6	Football helmet	$98.00
7	Football shoulder pad	$115.00
8	Softball	$20.00
9	Soccer ball	$26.00

In this diagram, you can see how a database is divided into tables of data where the information relates to each other. The tables are further divided into columns, which are made up of the fields that end up being pulled into your report.

Types of Joins

Inner (Direct) Join -

The fields that are returned from an inner join will be the fields from the two tables when the value in a row for the first table also appears in a row for the second table. So the data you get using the inner join will exclude values that do not exist in both of your data sources. In the Venn diagram, the information that would be returned is represented by the dark green overlap of the two circles



From our database example above, let's say the business needed a report that shows information on the purchases their returning customers make. The information on the customer, such as their address, is stored in the customers table. So we are using the customers table to pull the field for the customer's address into the report. The information on the items and quantity purchased are in the purchases table, so we are using that table to get the fields for the purchase description in the report. This is why we need to join the tables.

If you used an inner join to join the customers table and the purchases table, your report would look something like this:

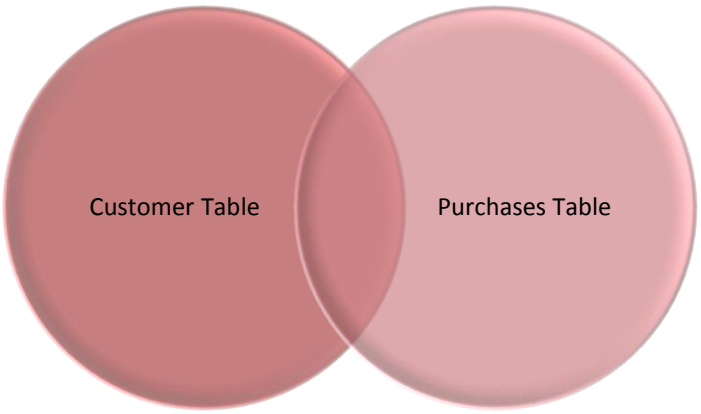
Customer Name	Address	Items Purchased	Qty
John Smith	563 Maple St.	Basketball	2
Kerry Jones	1212 Ivy Ln.	Football	1
Kerry Jones	1212 Ivy Ln.	Football Helmet	2
Reggie Brown	1623 Elm St.	Baseball	1
Reggie Brown	1623 Elm St.	Baseball hat	5

Compare this report to the database information above. You will notice that Sam Spade is not in this report even though he is a registered customer in their database. Also, the purchase that was made under guest in the purchase table is not showing in this report. This is because the results that are returned by an inner join are limited to information that exists in both tables, as represented by the dark green overlap in the Venn diagram above.

The results from the inner join may be exactly what you are intending to show. If the aim of your report was to show purchases made by registered customers, you would not want to include the purchase made by guest in the report, and you would not need to see that Sam Spade had no purchases. If, however, you were trying to build a report that showed all of the registered customer's purchase activity, you may want to see Sam Spade in your report, and see the null value in the purchase column. This is how the join can make a big difference in the success of your report.

Left (First Exists) -

A left join (left outer join) returns all rows from the left table, whether the joined columns have a match or not. Remember that the first table you select will always be your left table. A field will display a null value if the corresponding table does not contain a matching row. Since the left outer join returns all rows from the left table as well as the matching rows from the right table, if the field from the right table has no match, it will not be returned. This can be helpful in identifying when information is missing from records in the right table.



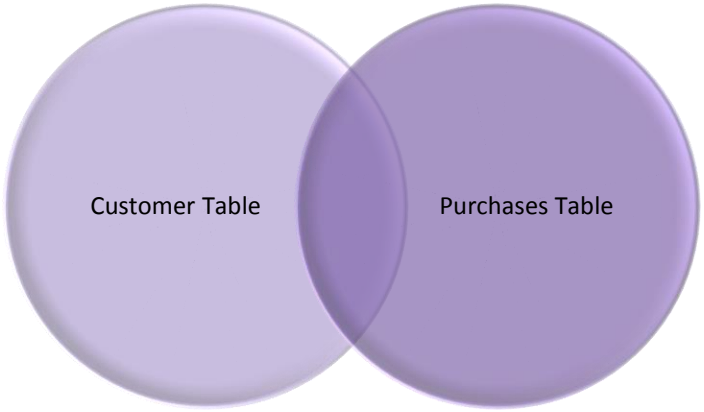
If we take the example report we used in the inner join description, and all we do to the report is apply the left join, the report would different data.

Customer Name	Address	Items Purchased	Qty
John Smith	563 Maple St.	Basketball	2
Kerry Jones	1212 Ivy Ln.	Football	1
Kerry Jones	1212 Ivy Ln.	Football Helmet	2
Reggie Brown	1623 Elm St.	Baseball	2
Reggie Brown	1623 Elm St.	Baseball hat	6
Sam Spade	909 Poplar Rd.	*Null*	*Null*

Now, using the left join, Sam Spade is in the report, and in the items purchased column and the quantity column there are null values. This is because Sam Spade's information is part of the customer's table, which is the left table in this example, and the left join will return all of the data from the left table.

Carefully considering the data that is relevant to your report will help you make the decision of which join to use.

Right Join -



A right join (also known as a right outer join) returns all rows from the right table in the right outer join clause, whether the joined columns match or not. A field in a result row displays a null if the corresponding table does not contain a matching row. If you use the right join (right outer join), the results that will be returned will be everything from the second data table, and the values that match from the first data table.

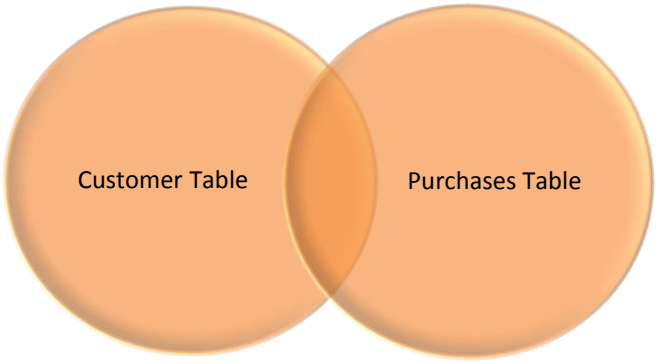
Like the left join returns results from the left table, the right join will return all of the data from the right table. Here is the same report used above, but this time with the right join:

Customer Name	Address	Items Purchased	Qty
John Smith	563 Maple St.	Basketball	2
Kerry Jones	1212 Ivy Ln.	Football	1
Kerry Jones	1212 Ivy Ln.	Football Helmet	2
Reggie Brown	1623 Elm St.	Baseball	2
Reggie Brown	1623 Elm St.	Baseball hat	6
Guest	*Null*	Football	2

We continue to see the results from registered customers who have made purchases, but now we see the purchase made by guest in the report. This is because the data for the guest purchase is stored in the purchases table, which was the right table in our example.

Full Join -

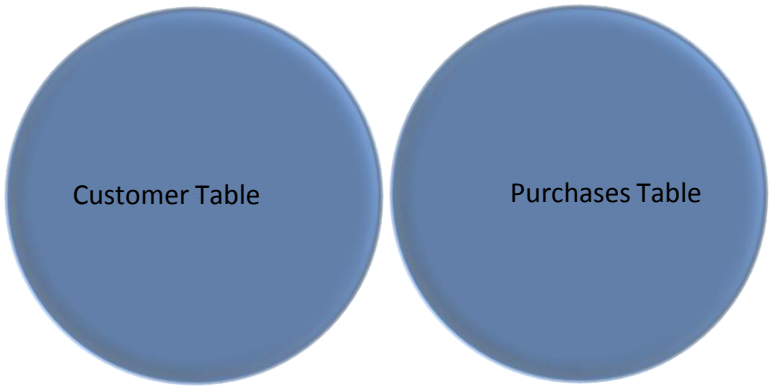
A full join (also known as a full outer join) returns all rows from both the right outer and left outer joins. A field in a result row displays a null if the corresponding table does not contain a matching row. In a full join, the full results from both tables will be returned, and the values within the records that do not match, will result in a null value.



Using the same report, if we change the left join to a full join, the results would change. Since the full join includes the data from the right table, the guest purchase that was made would also be returned in the results. And since it also includes all of the data from the left table, Sam Spade is also in the data returned. Here is the report when you use a full join:

Customer Name	Address	Items Purchased	Qty
John Smith	563 Maple St.	Basketball	2
Kerry Jones	1212 Ivy Ln.	Football	1
Kerry Jones	1212 Ivy Ln.	Football Helmet	2
Reggie Brown	1623 Elm St.	Baseball	2
Reggie Brown	1623 Elm St.	Baseball hat	6
Sam Spade	909 Poplar Rd.	*Null*	*Null*
Guest	*Null*	Football	2

Cross Join –
A cross join returns a result table where each row from the first table combines with each row from the second table. In the Venn diagram, both circles are dark in color since everything from both tables will be returned.



A cross join of recorded customers and products would look like this:

Customer Name	Product
John Smith	Basketball
John Smith	Football
John Smith	Baseball
Kerry Jones	Basketball
Kerry Jones	Football
Kerry Jones	Baseball
Reggie Brown	Basketball
Reggie Brown	Football
Reggie Brown	Baseball

If there were only these three customers and these three products in the database, this is what a cross join would produce.

Since the cross join returns so many results, you will likely never find a reason to use it when building your reports.