**Getting Data from Multiple Tables**

We can get results from multiple tables with either inner joins, outer joins, or unions. The most common join is the inner join. You join data from 2 tables in the FROM clause with the JOIN keyword. The ON keyword is used to define the join condition. Get all items ordered ever and sort them by id while listing their price :

**Inner Join**

SELECT item\_id, price

FROM item INNER JOIN sales\_item

ON item.id = sales\_item.item\_id

ORDER BY item\_id;

We use the join condition to find ids that are equal in the tables item and sales\_item. These joins are normally done using the primary and foreign keys in the tables as we did here. When we join tables while checking for equality between a common column this is called a equijoin.

You can define multiple join conditions with logical operators :

SELECT item\_id, price

FROM item INNER JOIN sales\_item

ON item.id = sales\_item.item\_id

AND price > 120.00

ORDER BY item\_id;

**Join 3 Tables**

Now let's join 3 tables. Get the orders, quantity and the total sale.

SELECT sales\_order.id, sales\_item.quantity, item.price,

(sales\_item.quantity \* item.price) AS Total

FROM sales\_order

JOIN sales\_item

ON sales\_item.sales\_order\_id = sales\_order.id

JOIN item

ON item.id = sales\_item.item\_id

ORDER BY sales\_order.id;

**Arithmetic Operators**

Other arithmetic operators include :

Addition : +

Subtraction : -

Division : /

Integer Division : DIV

Modulus : %

**Join with Where**

You can also define the join conditions using WHERE, but this is not considered to be a best practice.

SELECT item\_id, price

FROM item, sales\_item

WHERE item.id = sales\_item.item\_id

AND price > 120.00

ORDER BY item\_id;

**Outer Joins**

Outer joins return all of the rows from one of the tables being joined even if no matches are found.

A Left Outer Join returns all rows from the table being joined on the left. The Right Outer Join returns all rows from the table on the right. It's common practice to avoid Right Outer joins though.

Here I'll get product information from 2 tables

SELECT name, supplier, price

FROM product LEFT JOIN item

ON item.product\_id = product.id

ORDER BY name;

**Cross Joins**

Cross joins include data from each row in both tables. I'll grab information from the item and sales\_item table. This will produce many results. Since there are no join conditions in a Cross Join you will rarely use them.

SELECT sales\_order\_id, quantity, product\_id

FROM item CROSS JOIN sales\_item

ORDER BY sales\_order\_id;

**Unions & Extract**

Unions combine the results of 2 or more select statements in one result. Each result must return the same number of columns and data in each column must have the same data type.

Let's say we want to send birthday cards to all customers and sales persons for the month of December we could do this. Always put the Order By statement last. The column names are taken from those provided in the 1st select statement. (We use Extract to get just the month from the birth date)

SELECT first\_name, last\_name, street, city, zip, birth\_date

FROM customer

WHERE EXTRACT(MONTH FROM birth\_date) = 12

UNION

SELECT first\_name, last\_name, street, city, zip, birth\_date

FROM sales\_person

WHERE EXTRACT(MONTH FROM birth\_date) = 12

ORDER BY birth\_date;