

INTRODUCTION : MY SQL DEMOS WINTER 2017

INNER JOIN	Returns the rows present in both the table.
	<pre>SELECT spl_2016.deweyClass.deweyClass , spl_2016.itemToBib.itemNumber FROM spl_2016.deweyClass INNER JOIN spl_2016.itemToBib ON spl_2016.deweyClass.bibNumber = spl_2016.itemToBib.bibNumber;</pre>
LEFT JOIN	Returns all rows values present in table1 along with common rows between the two tables.
	<pre>SELECT spl_2016.deweyClass.deweyClass , spl_2016.itemToBib.itemNumber FROM spl_2016.deweyClass LEFT JOIN spl_2016.itemToBib ON spl_2016.deweyClass.bibNumber = spl_2016.itemToBib.bibNumber;</pre>
RIGHT JOIN	Returns all rows values present in table2 along with common rows between the two tables.
	<pre>SELECT spl_2016.deweyClass.deweyClass , spl_2016.itemToBib.itemNumber FROM spl_2016.deweyClass RIGHT JOIN spl_2016.itemToBib ON spl_2016.deweyClass.bibNumber = spl_2016.itemToBib.bibNumber;</pre>
INNER JOIN FOR THREE TABLES	Get the common rows in three tables.
	<pre>SELECT spl_2016.callNumber.callNumber ,spl_2016.collectionCode.itemNumber, spl_2016.itemToBib.bibNumber FROM spl_2016.callNumber,spl_2016.collectionCode,spl_2016.itemToBib WHERE spl_2016.callNumber.itemNumber=spl_2016.collectionCode.itemNumber AND spl_2016.collectionCode.itemNumber=spl_2016.itemToBib.itemNumber</pre>
GROUP BY , ORDER BY, FLOOR and COUNT	<p>GROUP BY : Group the rows into depending on a condition.</p> <p>ORDER BY : Order the rows in a descending or ascending fashion depending on a parameter.</p> <p>FLOOR : Round of to the closest integer.</p> <p>COUNT : Returns the number of rows in a table's column.</p>

	<p>EXAMPLE 1: SELECT FLOOR(deweyClass / 10) * 10 AS Dewey, COUNT(deweyClass) AS Number_of_Dewey_Items FROM spl_2016.deweyClass WHERE deweyClass > 0 GROUP BY FLOOR(deweyClass / 10) * 10 ORDER BY COUNT(deweyClass) DESC</p> <p>EXAMPLE 2: SELECT deweyClass AS Dewey, COUNT(deweyClass) AS Count FROM spl_2016.deweyClass WHERE deweyClass > 780 and deweyClass < 790 GROUP BY deweyClass ORDER BY deweyClass DESC LIMIT 200;</p>
CASE	<p>To get multidimensional data . Example here gives number of checkouts of every dewey class every month for the years between 2009-2015.</p>

```

SELECT
YEAR(checkout) AS Year,
MONTH(checkout) AS Month,
SUM(CASE
WHEN deweyClass >= 640 AND deweyClass < 641 THEN 1
ELSE 0
END) AS '640',
SUM(CASE
WHEN deweyClass >= 641 AND deweyClass < 642 THEN 1
ELSE 0
END) AS '641',
SUM(CASE
WHEN deweyClass >= 642 AND deweyClass < 643 THEN 1
ELSE 0
END) AS '642',
SUM(CASE
WHEN deweyClass >= 643 AND deweyClass < 644 THEN 1
ELSE 0
END) AS '643',
SUM(CASE
WHEN deweyClass >= 644 AND deweyClass < 645 THEN 1
ELSE 0
END) AS '644',
SUM(CASE
WHEN deweyClass >= 645 AND deweyClass < 646 THEN 1
ELSE 0
END) AS '645',
SUM(CASE
WHEN deweyClass >= 646 AND deweyClass < 647 THEN 1
ELSE 0
END) AS '646',
SUM(CASE
WHEN deweyClass >= 647 AND deweyClass < 648 THEN 1
ELSE 0
END) AS '647',
SUM(CASE
WHEN deweyClass >= 648 AND deweyClass < 649 THEN 1
ELSE 0
END) AS '648',
SUM(CASE
WHEN deweyClass >= 649 AND deweyClass < 650 THEN 1
ELSE 0
END) AS '649'
FROM
spl3._rawXmlDataCheckOuts
WHERE
itemtype = 'acbk'
AND YEAR(checkOut) >= '2009'
AND YEAR(checkOut) < '2015'
GROUP BY MONTH(checkOut) , YEAR(checkOut)
ORDER BY YEAR(checkOut) , MONTH(checkOut)

```

<p>AVG,MIN,MAX ,TIMESTAMP DIFF</p>	<p>AVG : returns average of the numeric column MIN : returns min of the numeric column MAX : return the max of the numeric column TIMESTAMPDIFF : returns the difference between two timestamps.</p>
	<pre> SELECT coutYear, COUNT(CASE WHEN Dewey > 100 AND Dewey<=200 THEN 1 END) as Class1_Count, AVG(CASE WHEN Dewey > 100 AND Dewey<=200 THEN ckoutDuration END) as Class1_Avg, MIN(CASE WHEN Dewey > 100 AND Dewey<=200 THEN ckoutDuration END) as Class1_Min, MAX(CASE WHEN Dewey > 100 AND Dewey<=200 THEN ckoutDuration END) as Class1_Max FROM (SELECT year(cout) as coutYear, DeweyClass as Dewey, (TIMESTAMPDIFF(minute, cout, cin))/(60*24) as ckoutDuration FROM spl_2016.inraw WHERE year(cout) >= "2010" AND (cin>cout)) AS new_table GROUP BY coutYear ORDER BY coutYear; </pre>