



“ACCOUNTING” ASSIGNMENT ON SQL

Chloë Bouton

HULT International Business School

Profit and Loss Statement for 2020 (% for changes are shown after, this query is used to create the views used after)

```

1 • SELECT Income_S.statement_section #Select the field we want to see in the P/L statement
2     ,Income_S.amount
3 FROM
4     (
5         SELECT z.statement_section
6             ,FORMAT((coalesce(ifnull(z.credit,0), ifnull(z.debit,0))) - (coalesce(ifnull(z.debit,0), ifnull(z.credit,0))),2) AS amount
7         FROM
8             (
9                 SELECT ss.statement_section
10                    ,jeli.debit
11                    ,jeli.credit
12                FROM statement_section AS ss # Join all the table together to obtain all the datas needed (amounts, sections, dates etc.)
13                INNER JOIN account AS a
14                    ON ss.statement_section_id = a.profit_loss_section_id
15                INNER JOIN journal_entry_line_item AS jeli
16                    ON a.account_id = jeli.account_id
17                INNER JOIN journal_entry AS je
18                    ON jeli.journal_entry_id = je.journal_entry_id
19                WHERE ss.company_id = 1 # Filter to only get the datas for company 1
20                  AND YEAR(je.entry_date) = 2020 # Filter to only get the datas from 2020
21                  AND je.cancelled = 0 # Exclude the entry that have been canceled
22                GROUP BY ss.statement_section # Group by statement section to put everything under one only statement (Current Assets et
23                ORDER BY ss.statement_section_id # Order by statement section id to have revenue in first and the following in order
24            ) z
25     ) Income_S # Naming the subquery
26
27 UNION # Union to get all the value (All statement sections and Net Income) following each others
28
29 (
30     SELECT 'NET INCOME' AS statement_section # Get the total net income at the end of the P/L statement
31     , (
32         SELECT FORMAT((ifnull(SUM(z.credit),0) - ifnull(SUM(z.debit),0)),2) AS amount # Calculation of Net Income with 2 decimals
33     FROM
34         (
35             SELECT ss.statement_section # From the same subquery above
36                 ,jeli.debit
37                 ,jeli.credit
38             FROM statement_section AS ss
39             INNER JOIN account AS a
40                 ON ss.statement_section_id = a.profit_loss_section_id
41             INNER JOIN journal_entry_line_item AS jeli
42                 ON a.account_id = jeli.account_id
43             INNER JOIN journal_entry AS je
44                 ON jeli.journal_entry_id = je.journal_entry_id
45             WHERE ss.company_id = 1
46                   AND YEAR(je.entry_date) = 2020
47                   AND je.cancelled = 0
48
49             GROUP BY ss.statement_section
50             ORDER BY ss.statement_section_id
51         ) z
52     )
53 )

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
statement_section	amount		
REVENUE	107,287.08		
COST OF GOODS AND SERVICES	-43,103.45		
SELLING EXPENSES	-6,896.55		
OTHER EXPENSES	-250.00		
OTHER INCOME	23,691.30		
NET INCOME	80,728.38		

Balance Sheet Statement for 2019 (% for changes are shown after, this query is used to create the views used after)

```

1  # Select fields with total debit, total credit, and total balance for each section with two decimals
2  # Ifnull means if value in field is null then 0
3  • SELECT
4      BS.statement_section,
5      FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2) AS Total_Balance
6  FROM
7      # From the subquery below which takes all the statements sections
8      (
9          SELECT ss.statement_section
10             , jeli.debit
11             , jeli.credit
12          FROM journal_entry_line_item AS jeli
13          INNER JOIN journal_entry AS je
14              ON jeli.journal_entry_id = je.journal_entry_id # Join both table on the same field on matches on journal entry ids
15          INNER JOIN account AS a
16              ON jeli.account_id = a.account_id # Join both table on the same field on matches on account ids
17          INNER JOIN statement_section AS ss
18              ON ss.statement_section_id = a.balance_sheet_section_id # Join both table on the same field on matches on B/S section ids
19
20          WHERE (a.company_id = 1 # Show results only for company 1
21                AND (ss.is_balance_sheet_section <> 0 # Exclude the datas not concerning the B/S
22                  AND (YEAR(je.entry_date) = 2019 # Filter for the year 2019 (2020 Missing datas)
23                    AND je.cancelled = 0))) # Exclude the entry that have been canceled
24          ORDER BY je.entry_date DESC # Order by most recent date first
25      ) AS BS # Naming the subquery
26
27      GROUP BY BS.statement_section # Group by statement section to put everything under one only statement (Current Assets etc.)
28
29      UNION # Union to get all the value (All statement sections and totals) following each others
30
31      # Calculation of Total Liabilities and Equity
32      SELECT 'TOTAL PASSIVE' AS statement_section # This is going to be used as a Title
33      , (SELECT FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2) # Takes the subtraction of debit by credit
34          FROM ( #for liabilities (see subquery below)
35
36              SELECT ss.statement_section
37                 , jeli.debit
38                 , jeli.credit
39              FROM journal_entry_line_item AS jeli # Join all the table together to obtain all the datas needed
40              INNER JOIN journal_entry AS je
41                  ON jeli.journal_entry_id = je.journal_entry_id
42              INNER JOIN account AS a
43                  ON jeli.account_id = a.account_id
44              INNER JOIN statement_section AS ss
45                  ON ss.statement_section_id = a.balance_sheet_section_id
46
47              WHERE (a.company_id = 1 # Show results only for company 1
48                    AND (ss.is_balance_sheet_section <> 0 # Exclude the datas not concerning the B/S
49                  AND (YEAR(je.entry_date) = 2019 # Filter for the year 2019 (2020 Missing datas)
50                    AND je.cancelled = 0))) # Exclude the entry that have been canceled
51              ORDER BY je.entry_date DESC # Order by most recent date first

```

```

47         ORDER BY je.entry_date DESC           # Order by most recent date first
48     ) AS BS                                   # Calling the subquery BS
49     WHERE statement_section = 'CURRENT LIABILITIES' # Filter to only use the code for Current Liabilities ...
50     OR statement_section = 'EQUITY') AS Total_Balance # ... or Equity # Naming the subquery
51
52 UNION
53
54 # Used the exact same subquery than above but for Total Assets
55 SELECT 'TOTAL ACTIVE' AS statement_section
56     , (SELECT FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2)
57     FROM (
58         SELECT ss.statement_section
59             , jeli.debit
60             , jeli.credit
61         FROM journal_entry_line_item AS jeli
62         INNER JOIN journal_entry AS je
63
64             ON jeli.journal_entry_id = je.journal_entry_id
65         INNER JOIN account AS a
66             ON jeli.account_id = a.account_id
67         INNER JOIN statement_section AS ss
68             ON ss.statement_section_id = a.balance_sheet_section_id
69         WHERE (a.company_id = 1
70             AND (ss.is_balance_sheet_section <> 0
71             AND (YEAR(je.entry_date) = 2019
72             AND je.cancelled = 0)))
73         ORDER BY je.entry_date DESC
74     ) AS BS
75     WHERE statement_section = 'FIXED ASSETS'
76     OR statement_section = 'CURRENT ASSETS') AS Total_Balance;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	statement_section	Total_Balance			
▶	CURRENT ASSETS	388,131.97			
	CURRENT LIABILITIES	-255,870.61			
	EQUITY	-132,261.36			
	TOTAL PASSIVE	-388,131.97			
	TOTAL ACTIVE	388,131.97			

P/L Statement for 2020 with percentages difference from 2019 (see Exhibit for the views)

```

1  # Created 2 views, PL_Statement_2020_CB2 and PL_Statement_2019_CB2 to make it easier
2  # PL_Statement_2020_CB2 is profit and loss statement for 2020
3  # PL_Statement_2019_CB2 is profit and loss statement for 2019
4
5  • SELECT p.statement_section
6      , p.amount
7      , ifnull(concat(percentage_change, '%'), '100%') AS Difference_with_2019  # Adding % to the results and removing nulls
8  FROM
9  (
10     SELECT coalesce(PL20.statement_section, PL19.statement_section) AS statement_section # Joining statement sections into one field
11         , PL20.amount AS amount
12         , FORMAT((((PL20.amount - PL19.amount) / PL19.amount)*100),2) AS percentage_change # Calculating the % difference btwn 2020 and 2019
13     FROM PL_Statement_2020_CB2 AS PL20
14     LEFT JOIN PL_Statement_2019_CB2 AS PL19 # Left join to have all the datas
15         ON PL20.statement_section = PL19.statement_section
16 )p;

```

statement_section	amount	Difference_with_2019
REVENUE	107,287.08	0.00%
COST OF GOODS AND SERVICES	-43,103.45	-50.00%
SELLING EXPENSES	-6,896.55	-53.85%
OTHER EXPENSES	-250.00	-0.00%
OTHER INCOME	23,691.30	100%
NET INCOME	80,728.38	1,042.86%

B/S Statement for 2019 with percentages difference from 2018 (Not enough data for 2020)

(See Exhibit for the views)

```

1  # Created 2 views, BS_2019_CB and BS_2018_CB to make it easier
2  # BS_2018_CB is B/S statement for 2018
3  # BS_2019_CB is B/S statement for 2019
4
5  • SELECT b.statement_section
6      , b.Total_Balance
7      , ifnull(concat(percentage_change, '%'), '100%') AS Difference_with_2018  # Adding % to the results and removing nulls
8  FROM
9  (
10     SELECT coalesce(BS19.statement_section, BS18.statement_section) AS statement_section # Joining statement sections into one field
11         , BS19.Total_Balance AS Total_Balance # Calculating the % difference btwn 2020 and 2019
12         , FORMAT((((BS19.Total_Balance - BS18.Total_Balance) / BS18.Total_Balance) * 100),2) AS percentage_change
13     FROM BS_2019_CB AS BS19
14     LEFT JOIN BS_2018_CB AS BS18 # Left join to have all the datas
15         ON BS19.statement_section = BS18.statement_section
16 )b;

```

statement_section	Total_Balance	Difference_with_2018
CURRENT ASSETS	388,131.97	18.65%
CURRENT LIABILITIES	-255,870.61	-572.22%
EQUITY	-132,261.36	-65.45%
TOTAL PASSIVE	-388,131.97	18.65%
TOTAL ACTIVE	388,131.97	18.65%

EXHIBITS

PL_Statement_2020_CB2 (YEAR(je.entry_date = 2020) was changed for 2019)

```

1 • CREATE VIEW PL_Statement_2020_CB2
2 AS
3 (SELECT z.statement_section AS statement_section
4     ,FORMAT((coalesce(ifnull(z.credit,0), ifnull(z.debit,0))) - (coalesce(ifnull(z.debit,0), ifnull(z.credit,0))),2) AS amount
5 FROM
6     (
7         SELECT ss.statement_section
8             ,jeli.debit
9             ,jeli.credit
10        FROM statement_section AS ss
11        INNER JOIN account AS a
12            ON ss.statement_section_id = a.profit_loss_section_id
13        INNER JOIN journal_entry_line_item AS jeli
14            ON a.account_id = jeli.account_id
15        INNER JOIN journal_entry AS je
16            ON jeli.journal_entry_id = je.journal_entry_id
17        WHERE ss.company_id = 1
18
19            AND YEAR(je.entry_date) = 2020
20            AND je.cancelled = 0
21        GROUP BY ss.statement_section
22        ORDER BY ss.statement_section_id
23    ) z
24 UNION
25
26 SELECT 'NET INCOME' AS statement_section
27     , (
28         SELECT FORMAT((ifnull(SUM(z.credit),0) - ifnull(SUM(z.debit),0)),2) AS amount
29         FROM
30             (
31                 SELECT ss.statement_section
32                     ,jeli.debit
33
34                     ,jeli.credit
35                 FROM statement_section AS ss
36                 INNER JOIN account AS a
37                     ON ss.statement_section_id = a.profit_loss_section_id
38                 INNER JOIN journal_entry_line_item AS jeli
39                     ON a.account_id = jeli.account_id
40                 INNER JOIN journal_entry AS je
41                     ON jeli.journal_entry_id = je.journal_entry_id
42                 WHERE ss.company_id = 1
43                     AND YEAR(je.entry_date) = 2020
44                     AND je.cancelled = 0
45                 GROUP BY ss.statement_section
46                 ORDER BY ss.statement_section_id
47             ) z
48     ) y
49 )

```

BS_2019_CB (YEAR(je.entry_date = 2019) was changed for 2018)

```
1 • CREATE VIEW BS_2019_CB
2 AS
3 (SELECT
4     BS.statement_section,
5     FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2) AS Total_Balance
6 FROM
7     (
8         SELECT ss.statement_section
9             , jeli.debit
10            , jeli.credit
11        FROM journal_entry_line_item AS jeli
12        INNER JOIN journal_entry AS je
13        ON jeli.journal_entry_id = je.journal_entry_id
14        INNER JOIN account AS a
15        ON jeli.account_id = a.account_id
16        INNER JOIN statement_section AS ss
17        ON ss.statement_section_id = a.balance_sheet_section_id
18
19        WHERE (a.company_id = 1
20            AND (ss.is_balance_sheet_section <> 0
21            AND (YEAR(je.entry_date) = 2019
22            AND je.cancelled = 0)))
23        ORDER BY je.entry_date DESC
24    ) AS BS
25
26    GROUP BY BS.statement_section
27
28 UNION
29
30 SELECT 'TOTAL PASSIVE' AS statement_section
31     , (SELECT FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2)
32     FROM (
33         SELECT ss.statement_section
34
35             , jeli.debit
36            , jeli.credit
37        FROM journal_entry_line_item AS jeli
38        INNER JOIN journal_entry AS je
39        ON jeli.journal_entry_id = je.journal_entry_id
40        INNER JOIN account AS a
41        ON jeli.account_id = a.account_id
42        INNER JOIN statement_section AS ss
43        ON ss.statement_section_id = a.balance_sheet_section_id
44        WHERE (a.company_id = 1
45            AND (ss.is_balance_sheet_section <> 0
46            AND (YEAR(je.entry_date) = 2019
47            AND je.cancelled = 0)))
48        ORDER BY je.entry_date DESC
49    ) AS BS
```

```

48         WHERE statement_section = "CURRENT LIABILITIES"
49         OR statement_section = "EQUITY") AS Total_Balance
50
51 UNION
52
53 SELECT 'TOTAL ACTIVE' AS statement_section
54     , (SELECT FORMAT((IFNULL(SUM(BS.debit),0) - IFNULL(SUM(BS.credit),0)),2)
55         FROM (
56             SELECT ss.statement_section
57                 , jeli.debit
58                 , jeli.credit
59             FROM journal_entry_line_item      AS jeli
60             INNER JOIN journal_entry          AS je
61             ON jeli.journal_entry_id = je.journal_entry_id
62             INNER JOIN account                AS a
63
64             ON jeli.account_id = a.account_id
65             INNER JOIN statement_section      AS ss
66             ON ss.statement_section_id = a.balance_sheet_section_id
67             WHERE (a.company_id = 1
68                 AND (ss.is_balance_sheet_section <> 0
69                     AND (YEAR(je.entry_date) = 2019
70                         AND je.cancelled = 0)))
71             ORDER BY je.entry_date DESC
72             ) AS BS
73         WHERE statement_section = "FIXED ASSETS"
74         OR statement_section = "CURRENT ASSETS") AS Total_Balance);

```