

Course Project

DeVry University

College of Engineering and Information Sciences

Course Number: MIS582
Course Project Deliverable: 6

Include screenshots of the code and result

Problem 1: Write a query to count the number of invoices.

SELECT COUNT(*) FROM INVOICE;

The screenshot displays the MySQL Workbench interface. The 'Query Editor' window contains the SQL query: `SELECT COUNT(*) FROM INVOICE;`. The 'Result Grid' window shows the output of the query, which is a single row with the value 1000. The 'Output' window shows the execution log, indicating that the query was executed successfully at 11:46:25 and returned 1 row(s).

Result Grid
COUNT(*)
1000

Output
1 11:46:25 SELECT COUNT(*) FROM INVOICE LIMIT 0, 1000

Problem 2: Generate a listing of all purchases made by the customers, using the output shown in the following as your guide. Sort the results by customer code, invoice number, and product description. You will need to join INVOICE, LINE, PRODUCT

SELECT CUS_CODE, INVOICE.INV_NUMBER, INV_DATE, P_DESCRIPT, LINE_UNITS, LINE_PRICE

**FROM INVOICE JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER JOIN PRODUCT
ON PRODUCT.P_CODE = LINE.P_CODE**

ORDER BY CUS_CODE, INVOICE.INV_NUMBER, P_DESCRIPT;

The screenshot displays the MySQL Workbench interface. The central pane shows a SQL query with three parts: a count of invoices, a join of invoice, line, and product tables, and a join of invoice and line tables. The results pane below shows a table with columns: CUS_CODE, INV_NUMBER, INV_DATE, P_DESCRIPT, LINE_UNITS, and LINE_PRICE. The table contains 18 rows of data. The bottom pane shows the execution output, indicating that the query was executed successfully and returned 18 rows.

CUS_CODE	INV_NUMBER	INV_DATE	P_DESCRIPT	LINE_UNITS	LINE_PRICE
10011	1002	2018-01-18 00:00:00	Rat-tail Rn, 1/8-in. fine	2.00	4.99
10011	1004	2018-01-17 00:00:00	Claw Hammer	2.00	9.95
10011	1008	2018-01-17 00:00:00	Rat-tail Rn, 1/8-in. fine	3.00	4.99
10011	1008	2018-01-17 00:00:00	Claw Hammer	1.00	9.95
10011	1008	2018-01-17 00:00:00	PVC pipe, 3/4-in., 8 ft	5.00	5.87
10011	1008	2018-01-17 00:00:00	Steel-nesting, 4-ftx18-in., 5" mesh	3.00	119.95
10012	1003	2018-01-18 00:00:00	7.25-in. prof. saw blade	5.00	14.99
10012	1003	2018-01-18 00:00:00	842 cordless drill, 12-in.	1.00	39.95
10012	1003	2018-01-18 00:00:00	Hnd. drill, 1/4-in., D-50	1.00	39.95
10014	1001	2018-01-18 00:00:00	7.25-in. prof. saw blade	1.00	14.99
10014	1001	2018-01-18 00:00:00	Claw Hammer	1.00	9.95
10014	1006	2018-01-17 00:00:00	1.25-in. metal screw, 25	3.00	6.99
10014	1006	2018-01-17 00:00:00	842 3/8-in. blade	1.00	109.95
10014	1006	2018-01-17 00:00:00	Claw Hammer	1.00	9.95
10014	1006	2018-01-17 00:00:00	Hnd. chain saw, 18 in.	1.00	256.99
10015	1007	2018-01-17 00:00:00	7.25-in. prof. saw blade	1.00	14.99
10015	1007	2018-01-17 00:00:00	Rat-tail Rn, 1/8-in. fine	1.00	4.99
10018	1005	2018-01-17 00:00:00	PVC pipe, 3/4-in., 8 ft	12.00	5.87

Problem 3 : Create a query to produce the total purchase per invoice, generating the results shown in the following *Figure*, sorted by invoice number. The invoice total is the sum of the product purchases in the `LINE` that corresponds to the `INVOICE`.

```
SELECT INV_NUMBER, SUM(LINE_UNITS * LINE_PRICE) AS "INVOICE TOTAL"
FROM LINE
GROUP BY INV_NUMBER
ORDER BY INV_NUMBER;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1  -- PROBLEM 1
2  SELECT COUNT(*) FROM INVOICE;
3
4
5  -- PROBLEM 2
6  SELECT CUS_CODE, INVOICE.INV_NUMBER, INV_DATE, P_DESCRPT, LINE_UNITS, LINE_PRICE
7  FROM INVOICE JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER JOIN PRODUCT ON PRODUCT.P_CODE = LINE.P_CODE
8  ORDER BY CUS_CODE, INVOICE.INV_NUMBER, P_DESCRPT;
9
10 -- PROBLEM 3
11 SELECT INV_NUMBER, SUM(LINE_UNITS * LINE_PRICE) AS "INVOICE TOTAL"
12 FROM LINE
13 GROUP BY INV_NUMBER
14 ORDER BY INV_NUMBER;
15
```

The Results window displays the output of the third query, showing a table with two columns: `INV_NUMBER` and `INVOICE TOTAL`.

INV_NUMBER	INVOICE TOTAL
1001	24.9400
1002	9.9800
1003	153.8300
1004	24.8700
1005	76.4600
1006	397.4300
1007	24.8700
1008	396.1300

The Output window shows the execution log:

Time	Action	Message	Duration / Fetch
16:45:25	SELECT COUNT(*) FROM INVOICE LIMIT 0, 1000	1 row(s) returned	0.076 sec / 0.000 sec
16:55:36	SELECT CUS_CODE, INVOICE.INV_NUMBER, INV_DATE, P_DESCRPT, LINE_UNITS, LINE_PRICE FROM INVOICE JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER JOIN PRODUCT ON PRODUCT.P_CODE = LINE.P_CODE ORDER BY CUS_CODE, INVOICE.INV_NUMBER, P_DESCRPT	18 row(s) returned	0.047 sec / 0.000 sec
16:55:44	SELECT INV_NUMBER, SUM(LINE_UNITS * LINE_PRICE) AS "INVOICE TOTAL" FROM LINE GROUP BY INV_NUMBER ORDER BY INV_NUMBER LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec