

## SIT103 – DATABASE FUNDAMENTALS

### SELECT QUERIES WITH JOIN 6.1P

1. Write a SQL statement to list invoices with their INV\_NUMBER, INV\_DATE along with the CUS\_FNAME and CUS\_LNAME of the customers they belong to.

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

```
1 create database sit103;
2 show tables;
3 show databases;
4 select INV.INV_NUMBER , INV.INV_DATE , CUS.CUS_FNAME , CUS.CUS_LNAME FROM IN
5 JOIN CUSTOMER CUS ON INV.CUS_CODE = CUS.CUS_CODE;
6
```

The Results window displays the following data:

INV_NUMBER	INV_DATE	CUS_FNAME	CUS_LNAME
1001	2018-01-16 00:00:00	Myron	Orlando
1002	2018-01-16 00:00:00	Leona	Dunne
1003	2018-01-16 00:00:00	Kathy	Smith
1004	2018-01-17 00:00:00	Leona	Dunne
1005	2018-01-17 00:00:00	Anne	Farris
1006	2018-01-17 00:00:00	Myron	Orlando
1007	2018-01-17 00:00:00	Amy	O'Brian
1008	2018-01-17 00:00:00	Leona	Dunne

2. Write a SQL statement to list V\_CODE and V\_NAME along with the number of products they supply (column name as 'NUM\_PRODS') and the average price of product they supply (column name as 'AVG\_PROD\_PRICE').

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

```
1 create database sit103;
2 show tables;
3 show databases;
4 select INV.INV_NUMBER , INV.INV_DATE , CUS.CUS_FNAME , CUS.CUS_LNAME FROM INVOICE INV
5 JOIN CUSTOMER CUS ON INV.CUS_CODE = CUS.CUS_CODE;
6 select V.V_CODE , V.V_NAME , COUNT(P.P_CODE) AS NUM_PRODS , AVG (P.P_PRICE) AS AVG_PROD_PRICE
7 FROM VENDOR V JOIN PRODUCT P ON V.V_CODE = P.V_CODE GROUP BY V.V_CODE , V.V_NAME order by V_CODE ASC;
```

The Results window displays the following data:

V_CODE	V_NAME	NUM_PRODS	AVG_PROD_PRICE
21225	Bryson, Inc.	2	8.470000
21231	DSE Supply	1	8.490000
21344	Gomez Bros.	3	12.460000
23119	Randssets Ltd.	2	41.970000
24288	ORDVA, Inc.	3	155.593333
25595	Rubicon Systems	3	89.630000

- Write a SQL statement to list P\_CODE and P\_DESCRIPT of all products along with their vendors' V\_CODE and V\_NAME if available. Note that your results must include all products regardless of whether vendor information is available or not. For products where vendor information is not available, V\_CODE and V\_NAME columns will be empty or NULL in the result (Hint: use outer join).

The screenshot shows MySQL Workbench with a SQL query in the 'Query 1' tab. The query is as follows:

```

1 show databases;
2
3 select INV.INV_NUMBER, INV.INV_DATE, CUS.CUS_FNAME, CUS.CUS_LNAME FROM INVOICE INV
4 JOIN CUSTOMER CUS ON INV.CUS_CODE = CUS.CUS_CODE;
5
6 SELECT V.V_CODE, V.V_NAME, COUNT(P.P_CODE) AS NUM_PROD, AVG(P.P_PRICE) AS AVG_PROD_PRICE
7 FROM VENDOR V JOIN PRODUCT P ON V.V_CODE = P.V_CODE GROUP BY V.V_CODE, V.V_NAME order by V_CODE ASC;
8
9 SELECT P.P_CODE, P.P_DESCRIPT, V.V_CODE, V.V_NAME
10 FROM PRODUCT P LEFT JOIN VENDOR V ON P.V_CODE = V.V_CODE;

```

The 'Result Grid' shows the results of the third query, displaying columns P\_CODE, P\_DESCRIPT, V\_CODE, and V\_NAME. The results are as follows:

P_CODE	P_DESCRIPT	V_CODE	V_NAME
11QER/31	Power painter, 15 psi., 3-nozzle	25595	Rubicon Systems
15-Q/P2	7.25-in. pair. saw blade	21344	Gomez Bros.
14-Q/L3	9.00-in. pair. saw blade	21344	Gomez Bros.
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	23119	Randslett Ltd.
1558-QW1	Hrd. cloth, 1/2-in., 3x50	23119	Randslett Ltd.
2232-QTY	BAD jigsaw, 12-in. blade	24288	ORDIVA, Inc.
2232-QWE	BAD jigsaw, 8-in. blade	24288	ORDIVA, Inc.
2238-QPD	BAD cordless drill, 1/2-in.	25595	Rubicon Systems
23109-HB	Clen hammer	21225	Bryson, Inc.
23114-AA	Sledge hammer, 12 lb.	21225	Bryson, Inc.
54778-ZT	Rat-tail file, 1/8-in. fine	21344	Gomez Bros.
89-WRE-Q	Hicut chain saw, 16 in.	24288	ORDIVA, Inc.

- Write a SQL statement to retrieve INV\_DATE and the list of products with P\_CODE and P\_DESCRIPT of the invoice with INV\_NUMBER = 1008 (Hint: you may have to join three tables)

The screenshot shows MySQL Workbench with a SQL query in the 'Query 1' tab. The query is as follows:

```

5 JOIN CUSTOMER CUS ON INV.CUS_CODE = CUS.CUS_CODE;
6 SELECT V.V_CODE, V.V_NAME, COUNT(P.P_CODE) AS NUM_PROD, AVG(P.P_PRICE) AS AVG_PROD_PRICE
7 FROM VENDOR V JOIN PRODUCT P ON V.V_CODE = P.V_CODE GROUP BY V.V_CODE, V.V_NAME order by V_CODE ASC;
8 SELECT P.P_CODE, P.P_DESCRIPT, V.V_CODE, V.V_NAME
9 FROM PRODUCT P LEFT JOIN VENDOR V ON P.V_CODE = V.V_CODE;
10 SELECT INV.INV_DATE, P.P_CODE, P.P_DESCRIPT FROM INVOICE INV JOIN LINE LN
11 ON INV.INV_NUMBER = LN.INV_NUMBER JOIN PRODUCT P ON LN.P_CODE = P.P_CODE WHERE INV.INV_NUMBER = 1008;

```

The 'Result Grid' shows the results of the fourth query, displaying columns INV\_DATE, P\_CODE, and P\_DESCRIPT. The results are as follows:

INV_DATE	P_CODE	P_DESCRIPT
2018-01-17 00:00:00	PVC2DRIT	PVC pipe, 3.5-in., 8 ft
2018-01-17 00:00:00	WRL3/TT3	Steel matting, 4x8x1/8", 5" mesh
2018-01-17 00:00:00	23109-HB	Clen hammer

- Write a SQL statement to list full names (by concatenating EMP\_FNAME and EMP\_LNAME separated by a space) of employees AS 'Employee' with their managers' full name (again by concatenating EMP\_FNAME and EMP\_LNAME separated by a space) AS 'Manager' (Hint: recursive/self join and string concatenation)

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

```

8 SELECT P.P_CODE, P.P_DESCRIPTION, V.V_CODE, V.V_NAME
9 FROM PRODUCT P LEFT JOIN VENDOR V ON P.V_CODE = V.V_CODE;
10 SELECT INV.INV_DATE, P.P_CODE, P.P_DESCRIPTION FROM INVOICE INV JOIN LINE LN
11 ON INV.INV_NUMBER = LN.INV_NUMBER JOIN PRODUCT P ON LN.P_CODE = P.P_CODE WHERE INV.INV_NUMBER = 1000;
12 SELECT CONCAT(E1.EMP_FNAME, ' ', E1.EMP_LNAME) AS Employee,
13        CONCAT(E2.EMP_FNAME, ' ', E2.EMP_LNAME) AS Manager
14 FROM EMP E1 LEFT JOIN EMP E2 ON E1.EMP_MGR = E2.EMP_MGR where E1.EMP_MGR is not null;

```

The Result Grid shows the following data:

Employee	Manager
Rhonda Lewis	George Kolmycz
Rhett Vandem	George Kolmycz
Anne Jones	George Kolmycz
John Lange	Robert Williams
Jeanine Smith	Robert Williams
Jorge Dante	Robert Williams
George Smith	Paul Wiesenbach
Leigha Genkad	Paul Wiesenbach
Rupert Wadlington	Robert Williams
Edward Johnson	George Kolmycz
Melanie Smythe	Robert Williams
Marie Brandon	Paul Wiesenbach

- Write a SQL statement to list P\_CODE and P\_DESCRIPTION of products that came in store (INDATE) in the month of February of 2018.

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

```

9 FROM PRODUCT P LEFT JOIN VENDOR V ON P.V_CODE = V.V_CODE;
10 SELECT INV.INV_DATE, P.P_CODE, P.P_DESCRIPTION FROM INVOICE INV JOIN LINE LN
11 ON INV.INV_NUMBER = LN.INV_NUMBER JOIN PRODUCT P ON LN.P_CODE = P.P_CODE WHERE INV.INV_NUMBER = 1000;
12 SELECT CONCAT(E1.EMP_FNAME, ' ', E1.EMP_LNAME) AS Employee,
13        CONCAT(E2.EMP_FNAME, ' ', E2.EMP_LNAME) AS Manager
14 FROM EMP E1 LEFT JOIN EMP E2 ON E1.EMP_MGR = E2.EMP_MGR where E1.EMP_MGR is not null;
15 SELECT P_CODE, P_DESCRIPTION FROM PRODUCT WHERE P.INDATE BETWEEN '2018-02-01' AND '2018-02-28';

```

The Result Grid shows the following data:

P_CODE	P_DESCRIPTION
89-WIRE-Q	Hout chain saw, 16 in.
PVC230RT	PVC pipe, 3.5-in., 8-ft
SW-23116	2.5-in. wd. screw, 50