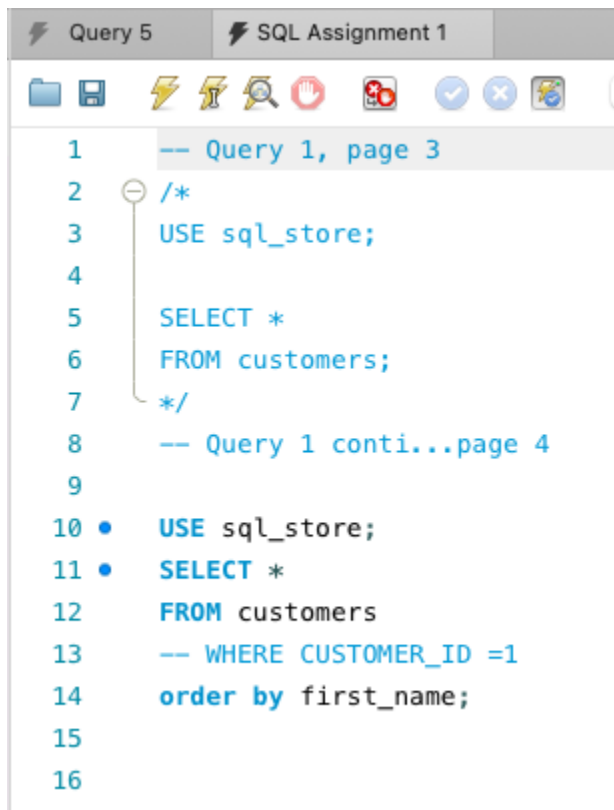


MY SQL Assignment Part 1

Page 2 – open ‘create-databases’ script. Done

Page 3- Query 1 and Page 4, Query 1 continued....



The screenshot shows a MySQL IDE window with two tabs: 'Query 5' and 'SQL Assignment 1'. The 'SQL Assignment 1' tab is active. The editor displays SQL code with line numbers 1 through 16. A comment on line 1 indicates the start of 'Query 1, page 3'. The code includes a multi-line comment block from line 2 to line 7, followed by a 'USE sql_store;' statement on line 3, a 'SELECT * FROM customers;' query on lines 5-6, and another 'USE sql_store;' statement on line 10. The query is continued on line 11 with 'SELECT * FROM customers' and line 12 with 'FROM customers'. Line 13 contains a comment '-- WHERE CUSTOMER_ID =1', and line 14 contains 'order by first_name;'. Lines 15 and 16 are empty.

```
1  -- Query 1, page 3
2  /*
3  USE sql_store;
4
5  SELECT *
6  FROM customers;
7  */
8  -- Query 1 conti...page 4
9
10 • USE sql_store;
11 • SELECT *
12 FROM customers
13 -- WHERE CUSTOMER_ID =1
14 order by first_name;
15
16
```

Page 5 – Query 2

The screenshot shows a SQL IDE interface with three tabs: "Query 5", "SQL Assignment 1", and "SQL Assignment 2*". The "Query 5" tab is active. The toolbar includes icons for file operations, execution, and a "Limit to 1000 rows" button. The query editor displays the following SQL code:

```
1  -- Query 2, page 5
2
3  • SELECT last_name, first_name, points, points+10
4     FROM CUSTOMERS;
5
```

Below the query editor, the "Result Grid" is visible, showing a table with 5 columns: last_name, first_name, points, points+10, and an empty column. The table contains 10 rows of data. The interface also includes a "Filter Rows" section with a search input and an "Export" button.

	last_name	first_name	points	points+10	
▶	MacCaffrey	Babara	2273	2283	
▶	Brushfield	Ines	947	957	
▶	Boagey	Freddi	2967	2977	
▶	Roseburgh	Ambur	457	467	
▶	Betchley	Clemmie	3675	3685	
▶	Twiddell	Elka	3073	3083	
▶	Dowson	Ilene	1672	1682	
▶	Naseby	Thacher	205	215	
▶	Rumgay	Romola	1486	1496	
▶	Mynett	Levy	796	806	

Task 1

12

```
7  -- Page 6 [(Points x 10) +100]
```

8




```
9  • SELECT last_name, first_name,points,(points*10 + 100)
```

```
10 FROM CUSTOMERS;
```

11

12

100% 16:10

Result Grid   Filter Rows: Export: 

	last_name	first_name	points	(points*10 + 100)	
▶	MacCaffrey	Babara	2273	22830	
▢	Brushfield	Ines	947	9570	
▢	Boagey	Freddi	2967	29770	
▢	Roseburgh	Ambur	457	4670	
▢	Betchley	Clemmie	3675	36850	
▢	Twiddell	Elka	3073	30830	
▢	Dowson	Ilene	1672	16820	
▢	Naseby	Thacher	205	2150	
▢	Rumgay	Romola	1486	14960	
▢	Mynett	Levy	796	8060	

Page 6

Task 1 (part 2)

```
12  -- Page 6 Create new column called 'discount_factor' which has formular (points+10)*100
13
14  • SELECT last_name, first_name, points, (points+10)*100 AS discount_factor
15  FROM CUSTOMERS;
16
17
```

100% 16:15

Result Grid Filter Rows: Search Export:

	last_name	first_name	points	discount_factor
▶	MacCaffrey	Babara	2273	228300
▶	Brushfield	Ines	947	95700
▶	Boagey	Freddi	2967	297700
▶	Roseburgh	Ambur	457	46700
▶	Betchley	Clemmie	3675	368500
▶	Twiddell	Elka	3073	308300
▶	Dowson	Ilene	1672	168200
▶	Naseby	Thacher	205	21500
▶	Rumgay	Romola	1486	149600
▶	Mynett	Levy	796	80600

Page 7

Task 2

Query 5 SQL Assignment 1 SQL Assignment 2*

Limit to 1000 rows

```
17
18  -- TASK 2 page 7, PPT 1
19  /*
20  Return all products. Columns shown are name, unit price and a new column called 'new price' which is = unit price *1.1.
21  i.e. You are increasing prices by 10%
22  New query returns all products with new price column
23  */
24
25  • USE sql_store;
26  • SELECT name, unit_price, unit_price*1.1 AS 'new_price'
27  FROM products;
28
29
```

100% 15:27

Result Grid Filter Rows: Search Export:

	name	unit_price	new_price
▶	Foam Dinner Plate	1.21	1.331
▶	Pork - Bacon,back Peameal	4.65	5.115
▶	Lettuce - Romaine, Heart	3.35	3.685
▶	Brocolinni - Gaylan, Chinese	4.53	4.983
▶	Sauce - Ranch Dressing	1.63	1.793
▶	Petit Baguette	2.39	2.629
▶	Sweet Pea Sprouts	3.29	3.619
▶	Island Oasis - Raspberry	0.74	0.814
▶	Longan	2.26	2.486
▶	Broom - Push	1.09	1.199

Page 8 Task 3

30 — Task 3

31 • SELECT*

32 FROM customers

```
33 WHERE birth_date> '1990-01-01';
```

34

100%	↕	31:33
------	---	-------

Result Grid

Filter Rows:

Search

Edit:





Export/Import:

[illegible]

Page 9 Task 4

```
37  -- Task 4
38  •  Use sql_inventory;
39  •  SELECT name, quantity_in_stock
40  FROM products
41  ORDER BY quantity_in_stock DESC
42  Limit 1
43  ;
44
```

100% 8:42



Result Grid   Filter Rows:

	name	quantity_in_sto...
▶	Sweet Pea Sprouts	98

Page 10 Task 5

```
45  -- Task 5
46  •  SELECT name, unit_price
47  FROM products
48  ORDER BY unit_price DESC
49  limit 1
50  .
```

100% 8:49



Result Grid   Filter Rows:

	name	unit_price
▶	Pork - Bacon,back Peameal	4.65

Page 11 Task 6

```
52  -- Task 6
53  • SELECT
54    first_name,
55    last_name,
56    address,
57    birth_date
58  FROM customers
59  ORDER BY birth_date
60  limit 1
61  ;
62
```

100% 2:61

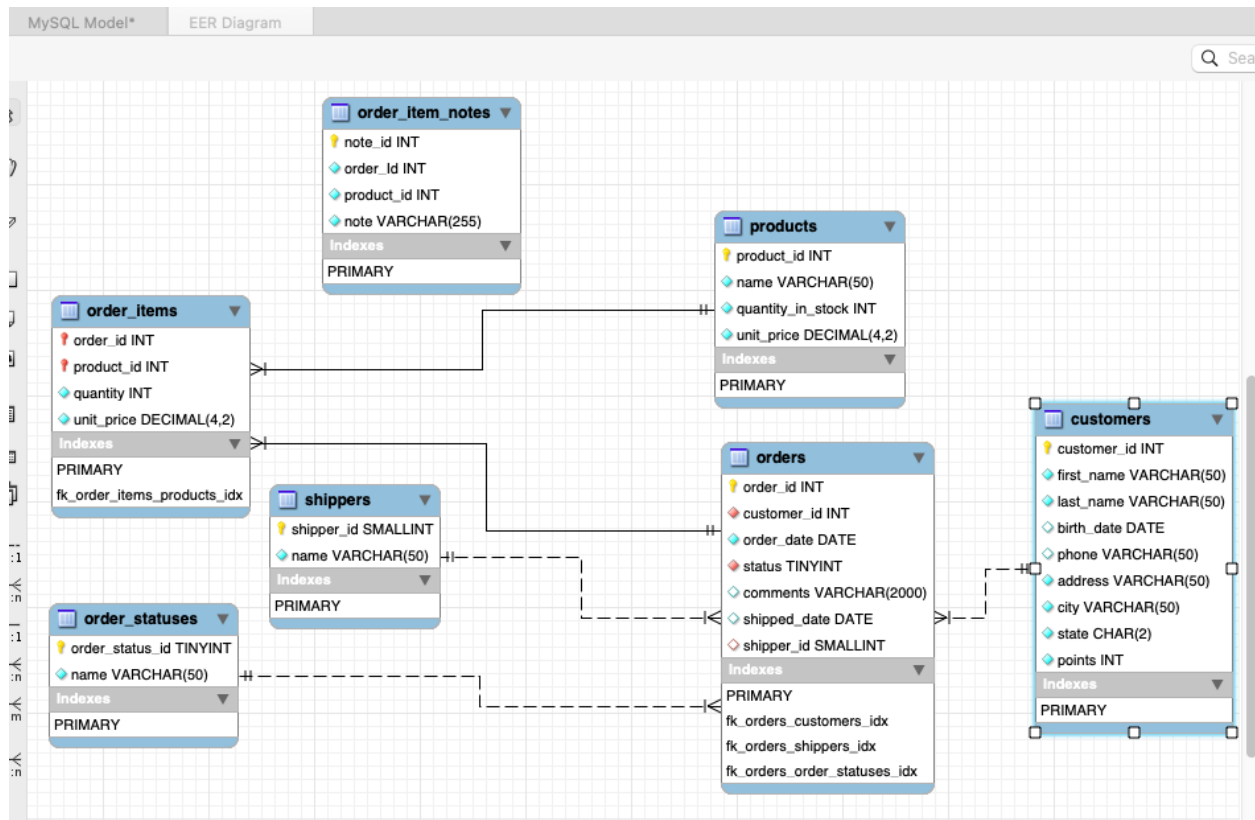
Result Grid   Filter Rows: [Export](#)

	first_name	last_name	address	birth_date
▶	Ilene	Dowson	50 Lillian Crossing	1964-08-30

Page 12 – Creating an EER Diagram

Page 13 – 17 Instructions

Page 18



Each box is a table i.e 'order_items' is a table

The lines that connect the boxes represent relationships between the tables i.e. order_items and products tables

Each bullet in the box is a column in the table.

If a column has a key icon next to it, it is a key.

Blue diamond icons mean no null values allowed (constraint exists)

White diamond icons means null values allowed (no constraints exists)

Red indicates its a foreign key

Primary key for that table is indicated by key icon.

Red Key icon means its a primary key (for that table) and a foreign key for another table.

CODE FROM SCRIPT (saved here for safety)

-- Query 1, page 3

/*

USE sql_store;

SELECT *

FROM customers;

*/

-- Query 1 conti...page 4

/*

USE sql_store;

SELECT *

FROM customers

-- WHERE CUSTOMER_ID =1

order by first_name;

*/

-- Query 2, page 5

/*

SELECT last_name, first_name, points, points+10

FROM CUSTOMERS;

*/

-- Page 6 [(Points x 10) +100]

/*

SELECT last_name, first_name, points, (points*10 + 100)

FROM CUSTOMERS;

*/

-- Page 6 Create new column called 'discount_factor' which has formular (points+10)*100

/*

SELECT last_name, first_name, points, (points+10)*100 AS discount_factor

FROM CUSTOMERS;

*/

-- TASK 2 page 7, PPT 1

/*

Return all products. Columns shown are name, unit price and a new column called 'new price' which is = unit price *1.1.

i.e. You are increasing prices by 10%

New query returns all products with new price column

*/

/*

USE sql_store;

SELECT name, unit_price, unit_price*1.1 AS 'new_price'

FROM products;

*/

-- Task 3

```
/*  
SELECT*  
FROM customers  
WHERE birth_date> '1990-01-01';  
*/
```

-- Task 4

```
Use sql_inventory;  
SELECT name, quantity_in_stock  
FROM products  
ORDER BY quantity_in_stock DESC  
Limit 1  
;
```

-- Task 5

```
SELECT name, unit_price  
FROM products  
ORDER BY unit_price DESC  
limit 1  
;
```

-- Task 6

```
SELECT  
first_name,  
last_name,  
address,
```

birth_date

FROM customers

ORDER BY birth_date

limit 1

;