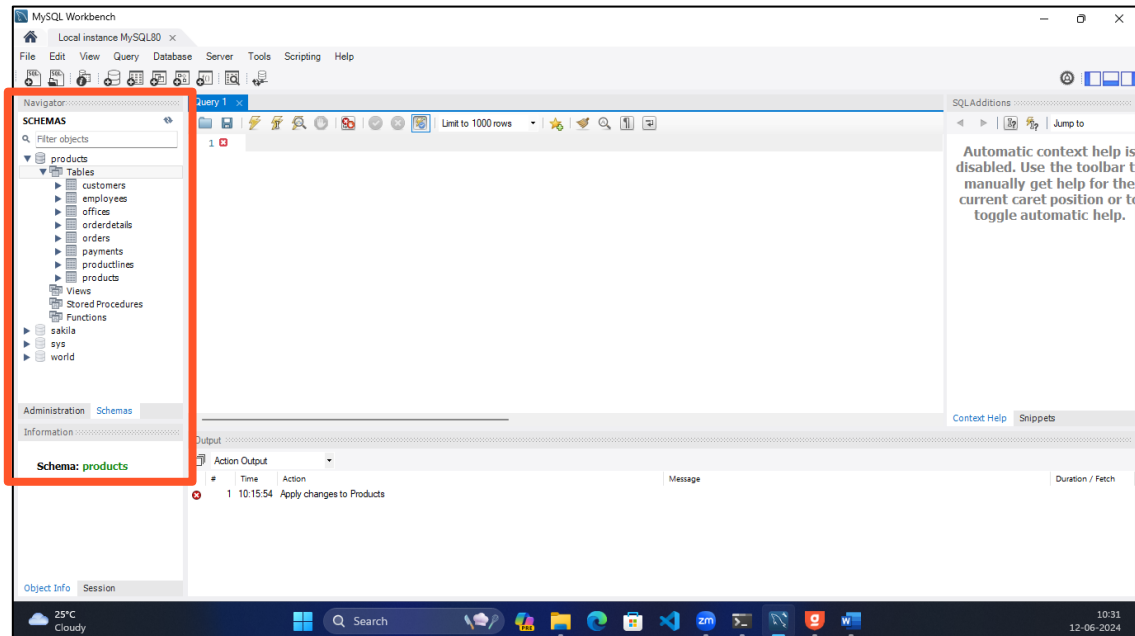


# MySQL Assignment 3

## Using WINDOWS

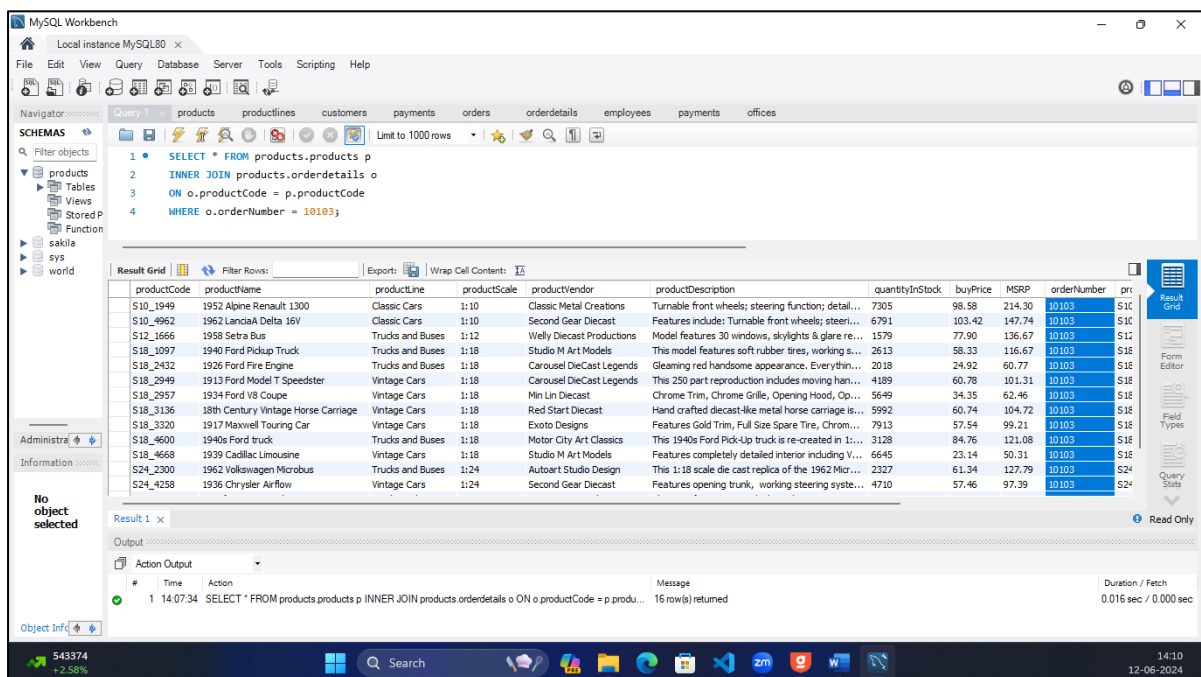
### TABLES CREATED SHOWN IN WORKBENCH



### Execution of all the queries

Q1. Write a query to retrieve the product details for order 10103.

```
SELECT * FROM products.products p
INNER JOIN products.orderdetails o
ON o.productCode = p.productCode
WHERE o.orderNumber = 10103;
```



Q2. Write a query to get the customer information for order 10127

```
SELECT * FROM products.customers c
INNER JOIN products.orders o
ON o.customerNumber = c.customerNumber
WHERE o.orderNumber = 10127;
```

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT * FROM products.customers c
2 INNER JOIN products.orders o
3 ON o.customerNumber = c.customerNumber
4 WHERE o.orderNumber = 10127;
```

The 'Result Grid' shows the output of the query, which is a single row of customer information for order 10127:

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit	orderNumber
151	Muscle Machine Inc	Young	Jeff	2125557413	4092 Furth Circle	Suite 400	NYC	NY	10022	USA	1286	138500.00	10127

The 'Action Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
8	14:15:31	SELECT * FROM products customers c LEFT JOIN products orders o ON o.customerNumber = c.custo...	1 row(s) returned	0.000 sec / 0.000 sec
9	14:15:55	SELECT * FROM products customers c INNER JOIN products orders o ON o.customerNumber = c.custo...	1 row(s) returned	0.000 sec / 0.000 sec

Q3. Write a query to retrieve the employee information for customer 166

```
SELECT e.employeeNumber, e.lastName, e.firstName, e.extension,
e.email, e.officeCode, e.reportsTo, e.jobTitle
FROM products.employees e
INNER JOIN products.customers c
ON e.employeeNumber = c.salesRepEmployeeNumber
WHERE c.customerNumber = 166;
```

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT e.employeeNumber, e.lastName, e.firstName, e.extension, e.email, e.officeCode, e.reportsTo, e.jobTitle
2 FROM products.employees e
3 INNER JOIN products.customers c
4 ON e.employeeNumber = c.salesRepEmployeeNumber
5 WHERE c.customerNumber = 166;
```

The 'Result Grid' shows the output of the query, which is a single row of employee information for customer 166:

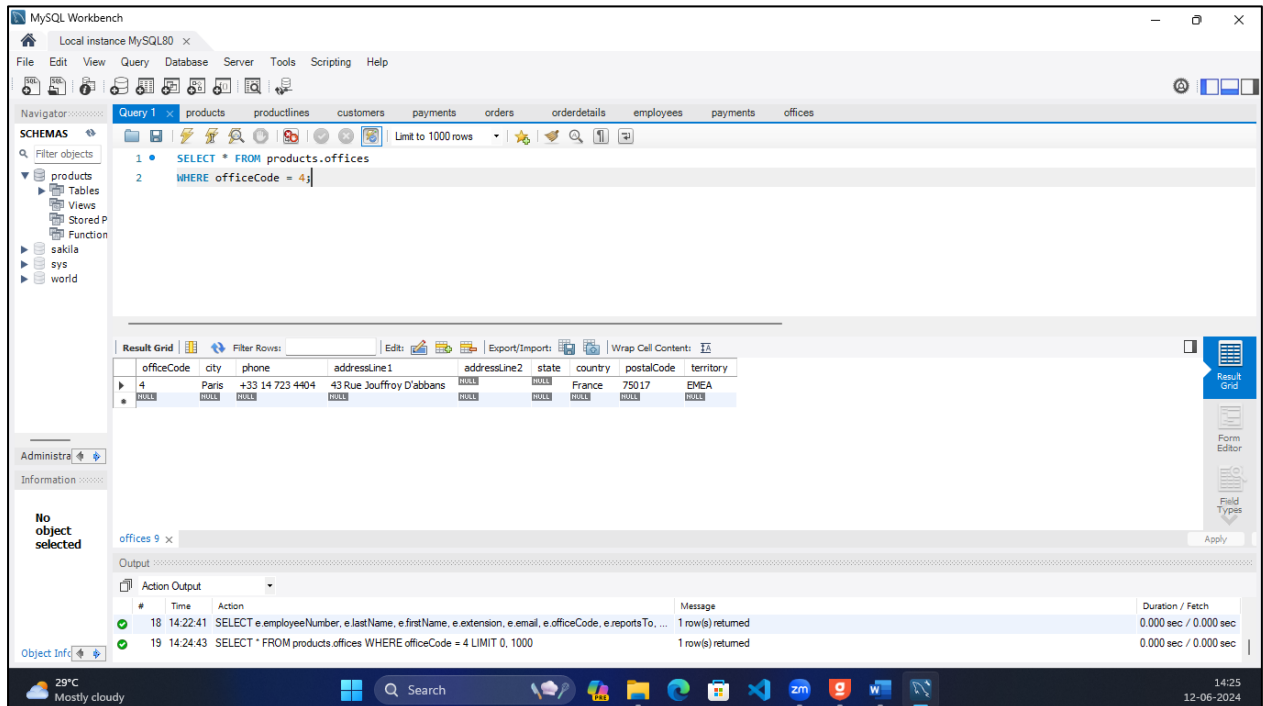
employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1612	Marsh	Peter	x102	pmarsh@classmodelcars.com	6	1088	Sales Rep

The 'Action Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
17	14:21:23	SELECT * FROM products employees e RIGHT JOIN products customers c ON e.employeeNumber = c...	1 row(s) returned	0.000 sec / 0.000 sec
18	14:22:41	SELECT e.employeeNumber, e.lastName, e.firstName, e.extension, e.email, e.officeCode, e.reportsTo, ...	1 row(s) returned	0.000 sec / 0.000 sec

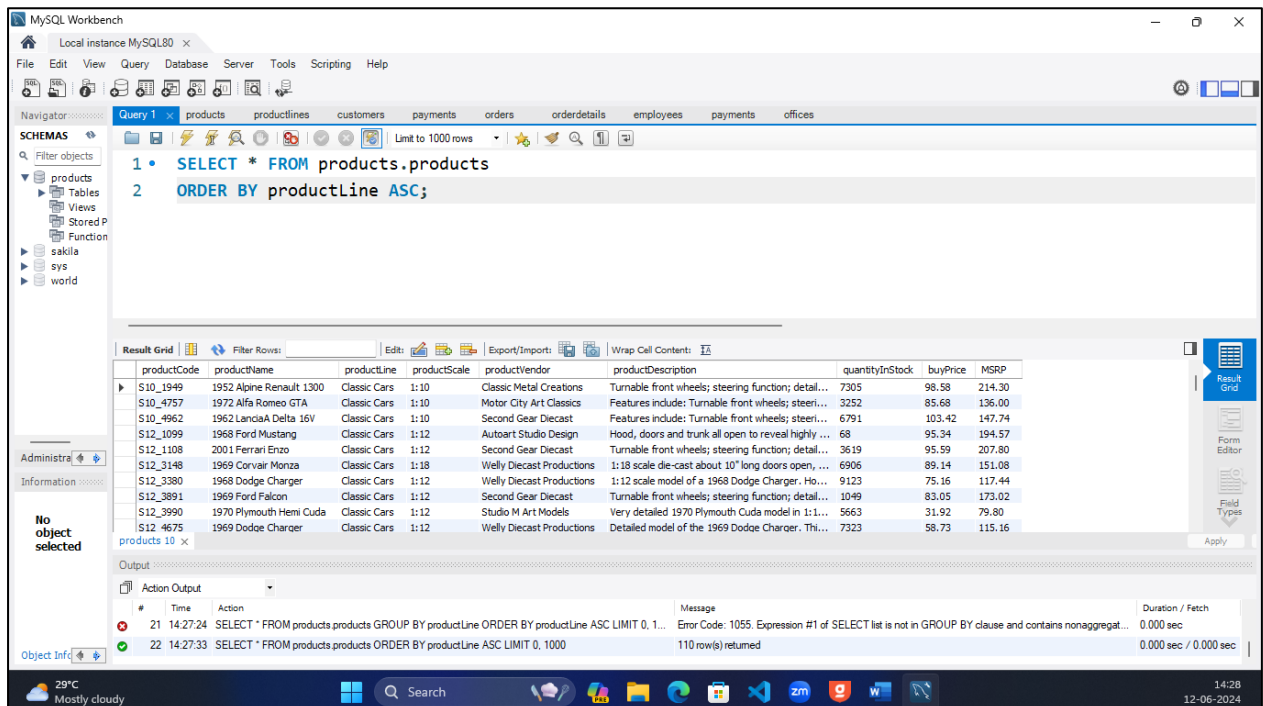
Q4. Write a query to get the official information for office code 4

```
SELECT * FROM products.offices
WHERE officeCode = 4;
```



Q5. Write a query to retrieve the product line for each product in an order.

```
SELECT * FROM products.products
ORDER BY productLine ASC;
```



Q6. Write a query to get the customer information and order status for all orders that contain products belonging to the 'Classic Cars' product line.

```
SELECT c.customerNumber, c.customerName, c.city, c.country,
c.creditLimit, c.state, c.phone, o.status as OrderStatus
FROM products.customers c
INNER JOIN products.orders o
ON o.customerNumber = c.customerNumber
INNER JOIN products.orderdetails od
ON od.orderNumber = o.orderNumber
INNER JOIN products.products p
ON od.productCode = p.productCode
WHERE p.productLine = 'Classic Cars';
```

MySQL Workbench - Local instance MySQL80

Query 1

```
1 SELECT c.customerNumber, c.customerName, c.city, c.country, c.creditLimit, c.state, c.phone, o.status as OrderStatus
2 FROM products.customers c
3 INNER JOIN products.orders o
4 ON o.customerNumber = c.customerNumber
5 INNER JOIN products.orderdetails od
6 ON od.orderNumber = o.orderNumber
7 INNER JOIN products.products p
8 ON od.productCode = p.productCode
9 WHERE p.productLine = 'Classic Cars';
```

Result Grid

	customerNumber	customerName	city	country	creditLimit	state	phone	OrderStatus
121	Baane Mini Imports	Stavern	Norway		81700.00		07-98 9555	Shipped
144	Volvo Model Replic...	Luleå	Sweden		53100.00		0921-12 3555	Shipped
498	Corrida Auto Repl...	Madrid	Spain		104600.00		(91) 555 22 82	Shipped
161	Technics Stores Inc.	Burlingame	USA		84600.00	CA	6505556809	Shipped
148	Dragon Souvenirs...	Singapore	Singapore		103800.00		+65 221 7555	Shipped
424	Classic Legends Inc.	NYC	USA		67500.00	NY	2125558493	Shipped
333	Australian Gift Net...	South Brisbane	Australia		51600.00	Queensland	61-7-3844-6555	Shipped
339	Classic Gift Ideas, ...	Philadelphia	USA		81100.00	PA	2155554695	Shipped
146	Saveley & Henrot...	Lyon	France		123900.00		78.32.5555	Shipped
202	Canadian Gift Exch...	Vancouver	Canada		90300.00	BC	(604) 555-3392	Shipped

Output

Action Output

Message

Duration / Fetch

0.000 sec / 0.000 sec

0.000 sec / 0.016 sec

Q7. Write a query to retrieve the payment details and customer details of customer number 103

```
SELECT c.customerNumber, c.customerName, c.city, c.state, c.country,
c.creditLimit, p.amount as PaymentAmount, p.checkNumber,
p.paymentDate
FROM products.customers c
INNER JOIN products.payments p
ON c.customerNumber = p.customerNumber
WHERE c.customerNumber = 103;
```

MySQL Workbench - Local instance MySQL80

Query 1

```
1 SELECT c.customerNumber, c.customerName, c.city, c.state, c.country, c.creditLimit, p.amount as PaymentAmount, p.checkNumber, p.paymentDate
2 FROM products.customers c
3 INNER JOIN products.payments p
4 ON c.customerNumber = p.customerNumber
5 WHERE c.customerNumber = 103;
```

Result Grid

	customerNumber	customerName	city	state	country	creditLimit	PaymentAmount	checkNumber	paymentDate
103	Atelier graphique	Nantes		France		21000.00	6066.78	HQ336336	2004-10-19
103	Atelier graphique	Nantes		France		21000.00	14571.44	JM555205	2003-06-05
103	Atelier graphique	Nantes		France		21000.00	1676.14	QM314933	2004-12-18

Output

Action Output

Message

Duration / Fetch

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

Q8. Write a query to get the orders and their corresponding payments to the same customer.

```
SELECT c.customerNumber, o.orderNumber, o.orderDate, o.requiredDate,
o.shippedDate, o.status, o.comments, p.amount as PaymentAmount,
p.checkNumber, p.paymentDate
FROM products.customers c
INNER JOIN products.payments p
ON c.customerNumber = p.customerNumber
INNER JOIN products.orders o
ON o.customerNumber = p.customerNumber
```

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

```
1 SELECT c.customerNumber, o.orderNumber, o.orderDate, o.requiredDate, o.shippedDate, o.status, o.comments, p.amount as PaymentAmount, p.checkNumber,
2 FROM products.customers c
3 INNER JOIN products.payments p
4 ON c.customerNumber = p.customerNumber
5 INNER JOIN products.orders o
6 ON o.customerNumber = p.customerNumber
```

The Results tab displays the query output as a table with 10 columns: customerNumber, orderNumber, orderDate, requiredDate, shippedDate, status, comments, PaymentAmount, checkNumber, and paymentDate. The table contains 16 rows of data.

customerNumber	orderNumber	orderDate	requiredDate	shippedDate	status	comments	PaymentAmount	checkNumber	paymentDate
363	10100	2003-01-06	2003-01-13	2003-01-10	Shipped		50799.69	HL575273	2004-11-17
363	10100	2003-01-06	2003-01-13	2003-01-10	Shipped		10223.83	IS232033	2003-01-16
363	10100	2003-01-06	2003-01-13	2003-01-10	Shipped		55425.77	PN238558	2003-12-05
128	10101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.	10549.01	DI925118	2003-01-28
128	10101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.	24101.81	FA465482	2003-10-18
128	10101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.	33820.62	FH668230	2004-03-24
128	10101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.	7466.32	JP383901	2004-11-18
181	10102	2003-01-10	2003-01-18	2003-01-14	Shipped		22602.36	CM564612	2004-04-25
181	10102	2003-01-10	2003-01-18	2003-01-14	Shipped		5494.78	GQ132144	2003-01-30
181	10102	2003-01-10	2003-01-18	2003-01-14	Shipped		44400.50	CH367219	2004-11-16

Q9. Write a query to retrieve the customers and their associated orders.

```
SELECT * FROM products.customers c
INNER JOIN products.orders o
ON o.customerNumber = c.customerNumber
```

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

```
1 SELECT * FROM products.customers c
2 INNER JOIN products.orders o
3 ON o.customerNumber = c.customerNumber
```

The Results tab displays the query output as a table with 12 columns: postalCode, country, salesRepEmployeeNumber, creditLimit, orderNumber, orderDate, requiredDate, shippedDate, status, comments, customerNumber, and orderNumber. The table contains 19 rows of data.

postalCode	country	salesRepEmployeeNumber	creditLimit	orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber	orderNumber
12005	USA	1216	114200.00	10100	2003-01-06	2003-01-13	2003-01-10	Shipped		363	10100
05028	Germany	1504	59700.00	10101	2003-01-09	2003-01-18	2003-01-11	Shipped	Check on availability.	128	10101
10022	USA	1286	76400.00	10102	2003-01-10	2003-01-18	2003-01-14	Shipped		181	10102
1110	Norway	1504	81700.00	10103	2003-01-29	2003-02-07	2003-02-02	Shipped		121	10103
38034	Spain	1370	227600.00	10104	2003-01-31	2003-02-09	2003-02-01	Shipped		141	10104
1734	Denmark	1401	83400.00	10105	2003-02-11	2003-02-21	2003-02-12	Shipped		145	10105
14100	Italy	1401	119600.00	10106	2003-02-17	2003-02-24	2003-02-21	Shipped		278	10106
10022	USA	1323	114900.00	10107	2003-02-24	2003-03-03	2003-02-26	Shipped	Difficult to negotiate with customer. We need m...	131	10107
1227MM	Philippines	1621	81500.00	10108	2003-03-03	2003-03-12	2003-03-08	Shipped		385	10108

Q10. Write a query to get the products and their corresponding product lines.

```
SELECT productCode, productName, productDescription, productLine
FROM products.products;
```

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 • SELECT productCode, productName, productDescription, productLine
2 FROM products.products;
```

The 'Result Grid' shows the results of the query, limited to 1000 rows. The columns are productCode, productName, productDescription, and productLine. The results are as follows:

productCode	productName	productDescription	productLine
S10_1678	1969 Harley Davidson Ultimate Chopper	This replica features working kickstand, front su...	Motorcycles
S10_1949	1952 Alpine Renault 1300	Turnable front wheels; steering function; detail...	Classic Cars
S10_2016	1996 Moto Guzzi 1100i	Official Moto Guzzi logos and insignias, saddle b...	Motorcycles
S10_4698	2003 Harley Davidson Eagle Drag Bike	Model features, official Harley Davidson logos a...	Motorcycles
S10_4757	1972 Alfa Romeo GTA	Features include: Turnable front wheels; steeri...	Classic Cars
S10_4962	1962 Lancia Delta 16V	Features include: Turnable front wheels; steeri...	Classic Cars
S12_1099	1968 Ford Mustang	Hood, doors and trunk all open to reveal highly ...	Classic Cars
S12_1108	2001 Ferrari Enzo	Turnable front wheels; steering function; detail...	Classic Cars
S12_1666	1958 Setra Bus	Model features 30 windows, skylights & glare re...	Trucks and Buses
S12_2823	2002 Suzuki XREO	Official logos and insignias, saddle bags located ...	Motorcycles

The 'Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
32	14:57:45	SELECT productCode, productName, productDescription, productLine FROM products.products LIMIT...	110 row(s) returned	0.000 sec / 0.000 sec
33	14:58:17	SELECT productCode, productName, productDescription, productLine FROM products.products LIMIT...	110 row(s) returned	0.000 sec / 0.000 sec

Q11. Write a query to retrieve the employees and their respective managers.

```
SELECT * FROM products.employees e1
JOIN (SELECT employeeNumber as ManagerNumber, CONCAT(e.lastName, ' ',
e.firstName) as ManagerName FROM products.employees e) AS e2
ON e2.ManagerNumber = e1.reportsTo
```

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 • SELECT * FROM products.employees e1
2 JOIN (SELECT employeeNumber as ManagerNumber, CONCAT(e.lastName, ' ', e.firstName) as ManagerName FROM products.employees e) AS e2
3 ON e2.ManagerNumber = e1.reportsTo
```

The 'Result Grid' shows the results of the query, limited to 1000 rows. The columns are employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle, ManagerNumber, and ManagerName. The results are as follows:

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle	ManagerNumber	ManagerName
1056	Patterson	Mary	x4611	mpatterson@classicmodelcars.com	1	1002	VP Sales	1002	Murphy Diane
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing	1002	Murphy Diane
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)	1056	Patterson Mary
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)	1056	Patterson Mary
1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	1056	Patterson Mary
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow Anthony
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	1143	Bow Anthony
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow Anthony
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow Anthony
1286	Tsenq	Foon Yue	x2248	ftsenq@classicmodelcars.com	3	1143	Sales Rep	1143	Bow Anthony

The 'Output' tab shows the execution details of the query:

#	Time	Action	Message	Duration / Fetch
56	15:19:45	SELECT * FROM products.employees e1 JOIN (SELECT employeeNumber as ManagerNumber, CONCAT(e.lastName, ' ', e.firstName) as ManagerName FROM products.employees e) AS e2	22 row(s) returned	0.000 sec / 0.000 sec
57	15:19:58	SELECT * FROM products.employees e1 JOIN (SELECT employeeNumber as ManagerNumber, CONCAT(e.lastName, ' ', e.firstName) as ManagerName FROM products.employees e) AS e2	22 row(s) returned	0.000 sec / 0.000 sec



Q12. Write a query to retrieve the customers, their orders, and the corresponding product details.

```
SELECT * FROM products.customers c
INNER JOIN products.orders o
ON c.customerNumber = o.customerNumber
INNER JOIN products.orderdetails od
ON o.orderNumber = od.orderNumber
INNER JOIN products.products p
ON od.productCode = p.productCode;
```

MySQL Workbench

Query 1

```
1 SELECT * FROM products.customers c
2 INNER JOIN products.orders o
3 ON c.customerNumber = o.customerNumber
4 INNER JOIN products.orderdetails od
5 ON o.orderNumber = od.orderNumber
6 INNER JOIN products.products p
7 ON od.productCode = p.productCode;
```

Result Grid

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit	orderNumber	orderDate	reship
363	Online Delect Creations Co.	Young	Dorothy	6035538647	2304 Long Airport Avenue		Nashua	NH	62005	USA	1216	114200.00	10100	2003-01-06	2003-01-06
363	Online Delect Creations Co.	Young	Dorothy	6035538647	2304 Long Airport Avenue		Nashua	NH	62005	USA	1216	114200.00	10100	2003-01-06	2003-01-06
363	Online Delect Creations Co.	Young	Dorothy	6035538647	2304 Long Airport Avenue		Nashua	NH	62005	USA	1216	114200.00	10100	2003-01-06	2003-01-06
128	Bauer See Auto. Co.	Kattel	Roland	+49 69 66 90 2555	Lyonerstr. 34		Frankfurt		60528	Germany	1504	99700.00	10101	2003-01-09	2003-01-09
128	Bauer See Auto. Co.	Kattel	Roland	+49 69 66 90 2555	Lyonerstr. 34		Frankfurt		60528	Germany	1504	99700.00	10101	2003-01-09	2003-01-09
128	Bauer See Auto. Co.	Kattel	Roland	+49 69 66 90 2555	Lyonerstr. 34		Frankfurt		60528	Germany	1504	99700.00	10101	2003-01-09	2003-01-09
181	Vladivone Inc.	Frisk	Michael	2125531000	2678 Kingston Rd.	Suite 101	NYC	NY	10022	USA	1286	79400.00	10102	2003-01-10	2003-01-10
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29
121	Baane Mini Imports	Berguffsen	Jonas	07-66 9535	Etling Skalkes gate 78		Stavem		4110	Norway	1504	81700.00	10103	2003-01-29	2003-01-29

Q13. Write a query to get the payment details, order details, and the associated products.

```
SELECT * FROM products.payments p
INNER JOIN products.orderdetails od
INNER JOIN products.orders o
ON p.customerNumber = o.customerNumber
INNER JOIN products.products pr
ON od.productCode = pr.productCode;
```

MySQL Workbench

Query 1

```
1 SELECT * FROM products.payments p
2 INNER JOIN products.orderdetails od
3 INNER JOIN products.orders o
4 ON p.customerNumber = o.customerNumber
5 INNER JOIN products.products pr
6 ON od.productCode = pr.productCode;
```

Result Grid

customerNumber	checkNumber	paymentDate	amount	orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	orderNumber	orderDate	requiredDate	ship
119	NG94694	2005-02-22	49523.67	10100	S18_1749	30	136.00	3	10425	2005-05-31	2005-06-07	10000
119	LN373947	2004-08-08	47924.19	10100	S18_1749	30	136.00	3	10425	2005-05-31	2005-06-07	10000
119	DB933704	2004-11-14	19501.82	10100	S18_1749	30	136.00	3	10425	2005-05-31	2005-06-07	10000
141	NU627706	2004-05-17	26155.91	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000
141	MF629602	2004-08-16	20009.53	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000
141	MC46946	2004-07-09	35420.74	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000
141	KT52578	2003-12-09	63843.55	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000
141	JN722810	2003-02-25	40206.20	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000
141	JN355280	2003-10-26	49539.37	10100	S18_1749	30	136.00	3	10424	2005-05-31	2005-06-08	10000

Output

Action Output

Message

Duration / Fetch

0.000 sec / 0.000 sec

0.000 sec / 0.000 sec

Q14. Write a query to retrieve the payment details and the customer information for the check number - JM555205

```
SELECT * FROM products.payments p
INNER JOIN products.customers c
ON p.customerNumber = c.customerNumber
WHERE p.checkNumber = 'JM555205'
```

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

```
1 SELECT * FROM products.payments p
2 INNER JOIN products.customers c
3 ON p.customerNumber = c.customerNumber
4 WHERE p.checkNumber = 'JM555205'
```

The Results tab displays the query output in a table format:

customerNumber	checkNumber	paymentDate	amount	customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2
103	JM555205	2003-06-05	14571.44	103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	

The bottom of the screenshot shows the 'Action Output' tab with the following message:

```
62 15:36:36 SELECT * FROM products.payments p INNER JOIN products.orderdetails od INNER JOIN pr... 1000 row(s) returned
63 15:40:01 SELECT * FROM products.payments p INNER JOIN products.customers c ON p.customerNu... 1 row(s) returned
```

Q15. Write a query to retrieve the orders and their corresponding customer and employee information for a canceled status.

```
SELECT * FROM products.orders o
INNER JOIN products.customers c
ON o.customerNumber = c.customerNumber
INNER JOIN products.employees e
ON e.employeeNumber = c.salesRepEmployeeNumber
WHERE o.status = 'Cancelled';
```

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

```
1 SELECT * FROM products.orders o
2 INNER JOIN products.customers c
3 ON o.customerNumber = c.customerNumber
4 INNER JOIN products.employees e
5 ON e.employeeNumber = c.salesRepEmployeeNumber
6 WHERE o.status = 'Cancelled';
```

The Results tab displays the query output in a table format:

orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber	customerNumber	customerName	contactLastName
10167	2003-10-23	2003-10-30		Cancelled	Customer called to cancel. The warehouse was ...	448	448	Scandinavian Gift Ideas	Larsson
10179	2003-11-11	2003-11-17	2003-11-13	Cancelled	Customer cancelled due to urgent budgeting iss...	496	496	Kelly's Gift Shop	Snowden
10248	2004-05-07	2004-05-14		Cancelled	Order was mistakenly placed. The warehouse n...	131	131	Land of Toys Inc.	Lee
10253	2004-06-01	2004-06-09	2004-06-02	Cancelled	Customer disputed the order and we agreed to ...	201	201	UK Collectables, Ltd.	Devon
10260	2004-06-16	2004-06-22		Cancelled	Customer heard complaints from their customer...	357	357	GiftsForHim.com	MacInlay
10262	2004-06-24	2004-07-01		Cancelled	This customer found a better offer from one of ...	141	141	Euro+ Shopping Channel	Freyre

The bottom of the screenshot shows the 'Action Output' tab with the following message:

```
64 15:43:35 SELECT * FROM products.orders o INNER JOIN products.customers c ON o.customerNumber... 6 row(s) returned
65 15:44:35 SELECT * FROM products.orders o INNER JOIN products.customers c ON o.customerNumber... 6 row(s) returned
```



Q16. Write a query to get the payments, order details, and associated product information for the payment date - 2004-12-17

```
SELECT * FROM products.payments p
INNER JOIN products.orderdetails od
INNER JOIN products.products pr
ON pr.productCode = od.productCode
WHERE p.paymentDate = '2004-12-17';
```

MySQL Workbench interface showing the execution of a query for Q16. The query is as follows:

```
1 SELECT * FROM products.payments p
2 INNER JOIN products.orderdetails od
3 INNER JOIN products.products pr
4 ON pr.productCode = od.productCode
5 WHERE p.paymentDate = '2004-12-17';
```

The result grid displays 10 rows of data:

customerNumber	checkNumber	paymentDate	amount	orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber	productCode	productName
447	OUS16561	2004-12-17	26304.13	10100	S18_1749	30	136.00	3	S18_1749	1917 Grand Touring Sedan
112	BO864823	2004-12-17	14191.12	10100	S18_1749	30	136.00	3	S18_1749	1917 Grand Touring Sedan
447	OUS16561	2004-12-17	26304.13	10100	S18_2248	50	55.09	2	S18_2248	1911 Ford Town Car
112	BO864823	2004-12-17	14191.12	10100	S18_2248	50	55.09	2	S18_2248	1911 Ford Town Car
447	OUS16561	2004-12-17	26304.13	10100	S18_4409	22	75.46	4	S18_4409	1932 Alfa Romeo 8C2300 Spider Sport
112	BO864823	2004-12-17	14191.12	10100	S18_4409	22	75.46	4	S18_4409	1932 Alfa Romeo 8C2300 Spider Sport
447	OUS16561	2004-12-17	26304.13	10100	S24_3969	49	35.29	1	S24_3969	1936 Mercedes Benz 500k Roadster
112	BO864823	2004-12-17	14191.12	10100	S24_3969	49	35.29	1	S24_3969	1936 Mercedes Benz 500k Roadster
447	OUS16561	2004-12-17	26304.13	10101	S18_2325	25	108.06	4	S18_2325	1932 Model A Ford 3-Coupe

Q17. Write a query to retrieve the products, order details, and corresponding customer information for customer 112

```
SELECT * FROM products.products pr
INNER JOIN products.orderdetails od
INNER JOIN products.orders o
ON od.productCode = pr.productCode
INNER JOIN products.customers c
ON c.customerNumber = o.customerNumber
WHERE c.customerNumber = 112;
```

MySQL Workbench interface showing the execution of a query for Q17. The query is as follows:

```
1 SELECT * FROM products.products pr
2 INNER JOIN products.orderdetails od
3 INNER JOIN products.orders o
4 ON od.productCode = pr.productCode
5 INNER JOIN products.customers c
6 ON c.customerNumber = o.customerNumber
7 WHERE c.customerNumber = 112;
```

The result grid displays 10 rows of data:

productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP	orderNumber
S18_1749	1917 Grand Touring Sedan	Vintage Cars	1:18	Welly Diecast Productions	This 1:18 scale replica of the 1917 Grand Touring Sedan	2724	86.70	170.00	10100
S18_1749	1917 Grand Touring Sedan	Vintage Cars	1:18	Welly Diecast Productions	This 1:18 scale replica of the 1917 Grand Touring Sedan	2724	86.70	170.00	10100
S18_1749	1917 Grand Touring Sedan	Vintage Cars	1:18	Welly Diecast Productions	This 1:18 scale replica of the 1917 Grand Touring Sedan	2724	86.70	170.00	10100
S18_2248	1911 Ford Town Car	Vintage Cars	1:18	Motor City Art Classics	Features opening hood, opening doors, opening windows	540	33.30	60.54	10100
S18_2248	1911 Ford Town Car	Vintage Cars	1:18	Motor City Art Classics	Features opening hood, opening doors, opening windows	540	33.30	60.54	10100
S18_2248	1911 Ford Town Car	Vintage Cars	1:18	Motor City Art Classics	Features opening hood, opening doors, opening windows	540	33.30	60.54	10100
S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars	1:18	Exoto Designs	This 1:18 scale precision die cast replica feature...	6553	43.26	92.03	10100
S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars	1:18	Exoto Designs	This 1:18 scale precision die cast replica feature...	6553	43.26	92.03	10100
S18_4409	1932 Alfa Romeo 8C2300 Spider Sport	Vintage Cars	1:18	Exoto Designs	This 1:18 scale precision die cast replica feature...	6553	43.26	92.03	10100

An error message is visible in the output pane: "Error Code: 1054. Unknown column 'od.customerNumber' in 'on clause'".

Q18. Write a query to retrieve the customers, their orders, and the associated product line information for the customers who are all from Boston.

```
SELECT pr.productLine, c.*, o.* FROM products.customers c
INNER JOIN products.orders o
ON o.customerNumber = c.customerNumber
INNER JOIN products.productlines pl
INNER JOIN products.orderdetails od
ON od.orderNumber = o.orderNumber
INNER JOIN products.products pr
ON pr.productCode = od.productCode
WHERE c.city = 'Boston';
```

MySQL Workbench interface showing the execution of a query. The query is displayed in the Query Editor, and the results are shown in the Result Grid. The query filters for customers in Boston and joins the customers, orders, product lines, and products tables.

productLine	customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode
Motorcycles	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Classic Cars	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Motorcycles	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Classic Cars	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Motorcycles	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Motorcycles	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Classic Cars	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Motorcycles	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Classic Cars	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003
Vintage Cars	362	Gifts4AllAges.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	51003

Q19. Write a query to get the employees, their respective managers, and the corresponding office details of the Sales Rep.

```
SELECT e1.*, e2.ManagerNumber, e2.ManagerName, o.*
FROM products.employees e1
JOIN (SELECT employeeNumber as ManagerNumber, CONCAT(e.lastName, ' ',
e.firstName) as ManagerName FROM products.employees e) AS e2
ON e2.ManagerNumber = e1.reportsTo
INNER JOIN products.offices o
ON e1.officeCode = o.officeCode;
```

MySQL Workbench interface showing the execution of a query. The query is displayed in the Query Editor, and the results are shown in the Result Grid. The query joins the employees table with a subquery that identifies managers and with the offices table.

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle	ManagerNumber	ManagerName	officeCode	city
1056	Patterson	Mary	x4611	mpatterson@classicmodelcars.com	1	1002	VP Sales				
1076	Pirrell	Jeff	x9273	jfirrell@classicmodelcars.com	1	1002	VP Marketing				
1143	Bow	Anthony	x4528	abow@classicmodelcars.com	1	1056	Sales Manager (NA)				
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep				
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep				
1188	Pirrell	Julie	x2173	jfirrell@classicmodelcars.com	2	1143	Sales Rep				
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep				
1286	Tseng	Foon Yue	x2248	ftseng@classicmodelcars.com	3	1143	Sales Rep				
1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep				

Q20. Write a query to retrieve the product lines, products, and the corresponding customer information for Vintage Cars.

```
SELECT pr.*,c.* From products.products pr
INNER JOIN products.customers c
INNER JOIN products.orders o
ON c.customerNumber = o.customerNumber
INNER JOIN products.orderdetails od
ON o.orderNumber = od.orderNumber
ON pr.productCode = od.productCode
WHERE pr.productLine = 'Vintage Cars';
```

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

```
1 SELECT pr.*,c.* From products.products pr
2 INNER JOIN products.customers c
3 INNER JOIN products.orders o
4 ON c.customerNumber = o.customerNumber
5 INNER JOIN products.orderdetails od
6 ON o.orderNumber = od.orderNumber
7 ON pr.productCode = od.productCode
8 WHERE pr.productLine = 'Vintage Cars';
```

The results are displayed in the Result Grid, showing 52 rows. The columns are: iuctCode, productName, productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP, customerNumber, and customerName. The results are filtered to show only rows where the productLine is 'Vintage Cars'.

iuctCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP	customerNumber	customerName
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	181	Vitachrome In...
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	129	Mini Wheels C...
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	114	Australian Co...
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	282	Souvenirs Ar...
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	487	Signal Collect...
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	321	Corporate Gift
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	278	Royall Gifts
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	124	Mini Gifts Dist
1342	1937 Lincoln Berline	Vintage Cars	1:18	Motor City Art Classics	Features opening engine cover, doors, trunk, a...	8693	60.62	102.74	471	Australian Co...

The Output pane shows the execution of the query, indicating that 657 row(s) were returned.