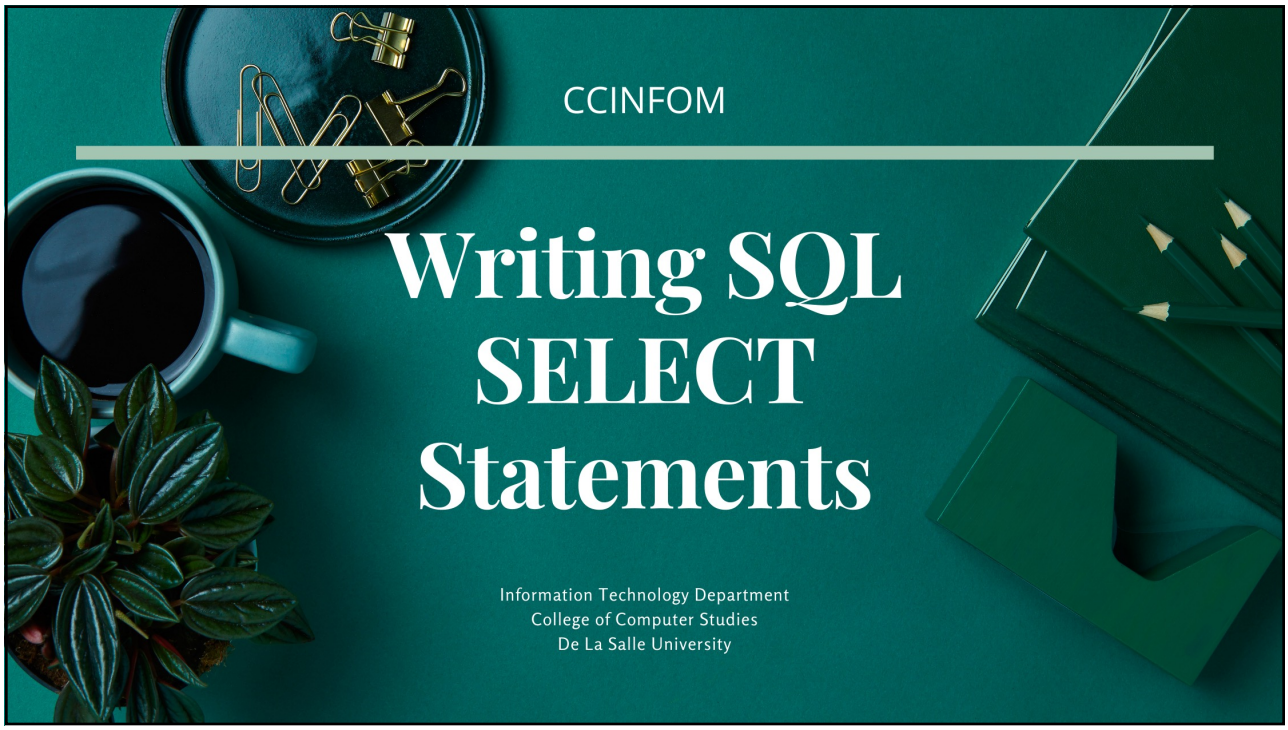


1

MySQL Script Files to be Used in this Session

- dbworld.sql
- ccinfomdemo.sql
- dbsales.sql

9

Thinking Discipline in Writing SQL Statements

Information Technology Department
De La Salle University

10

Sequence of Clauses in an SQL SELECT Statement

CLAUSE	DESCRIPTION
SELECT	What data will appear in the information result
FROM	Where will the data be coming from
WHERE	What conditions will be used to filter the data
ORDER BY	What fields will the information be sorted on

Information Technology Department
De La Salle University

11

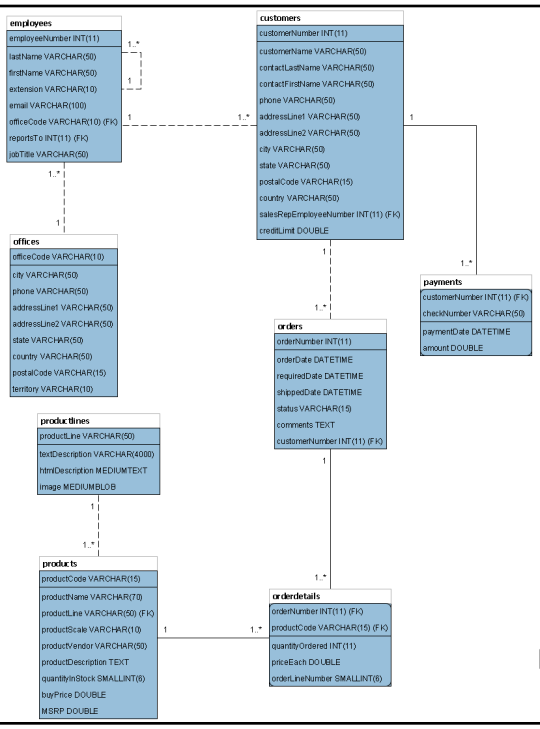
Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

Information Technology Department
De La Salle University

12

dbsales schema



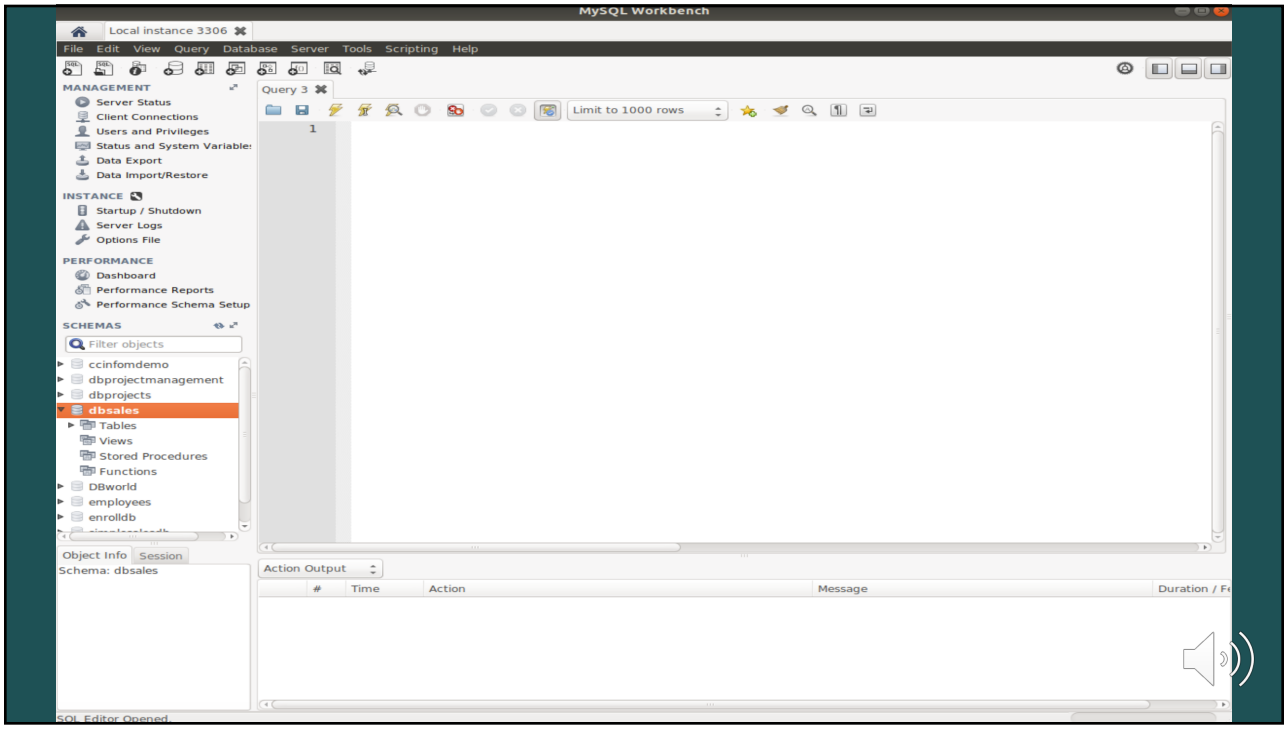
13

Practice Exercise

1. Create a new database using the script file "dbsales.sql".
2. Using the dbsales database/schema, write the SQL statement that will satisfy each data requirement in the succeeding slide.

Information Technology Department
De La Salle University

14



15

Practice Exercise

1. Generate a list of customers from Singapore. (Returns 3 rows)
2. Get the list of orders which were not fulfilled on time. (Returns 2 rows)
3. Get the total amount of payments made in 2003. (Returns 1 row)
4. Get the total number of customers per country. (Returns 27 rows)
5. Get the number of products per vendor. (Returns 13 rows)

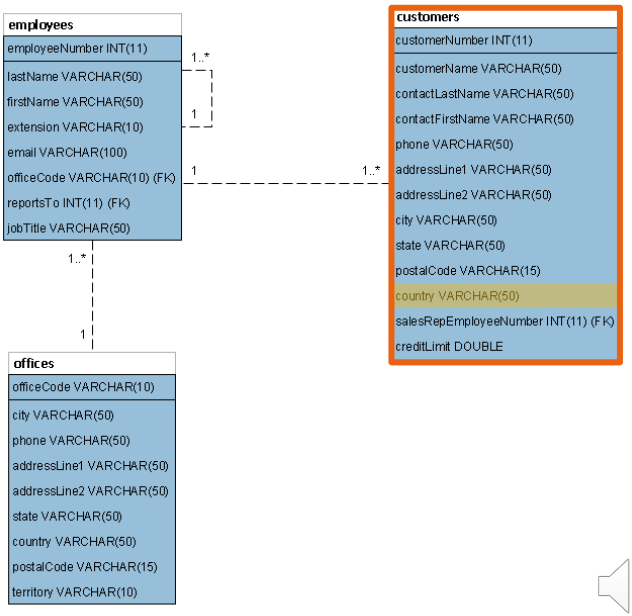
16

SQL Practice Exercises

1. Generate a list of customers from Singapore.

19

dbsales schema



20

SQL Practice Exercises

1. Generate a list of customers from Singapore.

```
SELECT      *
FROM        customers
WHERE       country = 'Singapore';
```


21

#	customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1
1	148	Dragon Souveniers, Ltd.	Natividad	Eric	+65 221 7555	Bronz Sok.
2	166	Handji Gifts& Co	Victorino	Wendy	+65 224 1555	106 Linden Road Sandow
3	206	Asian Shopping Network, Co	Walker	Brydey	+612 9411 1555	Suntec Tower Three
*	NULL	NULL	NULL	NULL	NULL	NULL

Returns 3 Rows

22

SQL Writing Practice Exercise

1. Generate a list of customers from Singapore.

```
SELECT      customernumber, customername, city
FROM        customers
WHERE       country = 'Singapore';
```

Information Technology Department
De La Salle University

23

SQL Writing Practice Exercise

1. Generate a list of customers from Singapore.

```
SELECT      c.customernumber, c.customername, c.city
FROM        customers c
WHERE       country = 'Singapore';
```

We use an alias to explicitly specify the table where a field would come from, especially in cases where two or more tables have field/s with the same name.

24

SQL Writing Practice Exercise

1. Generate a list of customers from Singapore.

```
SELECT      customernumber, customername,
            addressline1, addressline2, city
FROM        customers
WHERE       country = 'Singapore';
```

25

#	customernumber	customername	addressline1	addressline2	city
1	148	Dragon Souveniers, Ltd.	Bronz Sok.	Bronz Apt. 3/6 Tesvikiye	Singapore
2	166	Handji Gifts& Co	106 Linden Road Sandown	2nd Floor	Singapore
3	206	Asian Shopping Network, Co	Suntec Tower Three	8 Temasek	Singapore

Returns 3 Rows

26

SQL Writing Practice Exercise

1. Generate a list of customers from Singapore.

```
SELECT      customernumber, customername,
            addressline1, addressline2, city
FROM        customers
WHERE       country = 'Singapore'
ORDER BY    customername;
```

Information Technology Department
De La Salle University

27

SQL Writing Practice Exercise

1. Generate a list of customers from Singapore.

```
SELECT      customernumber, customername,
            addressline1, addressline2, city
FROM        customers
WHERE       country = 'Singapore'
ORDER BY    customername ASC;
```

ASC: ascending

DESC: descending

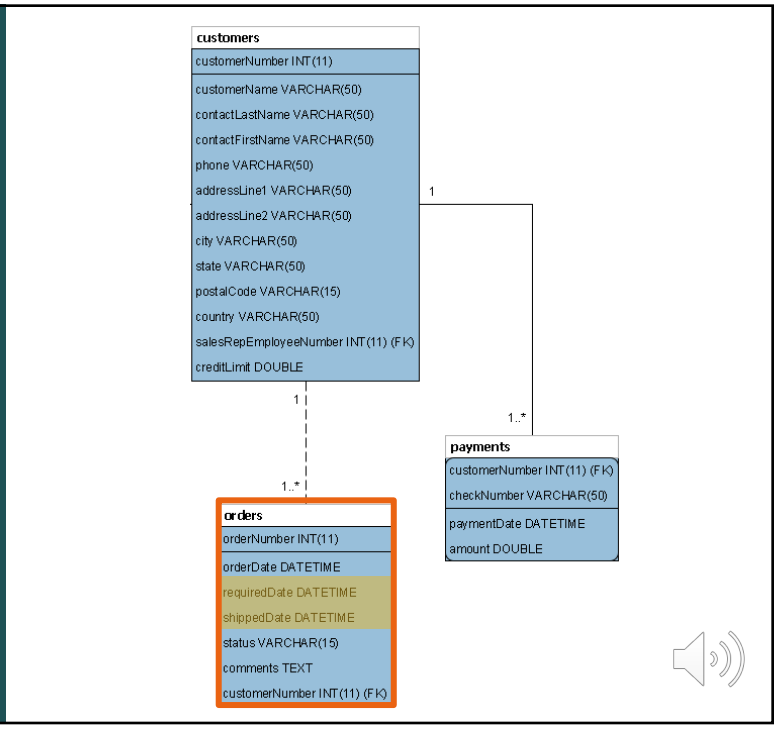
28

SQL Practice Exercises

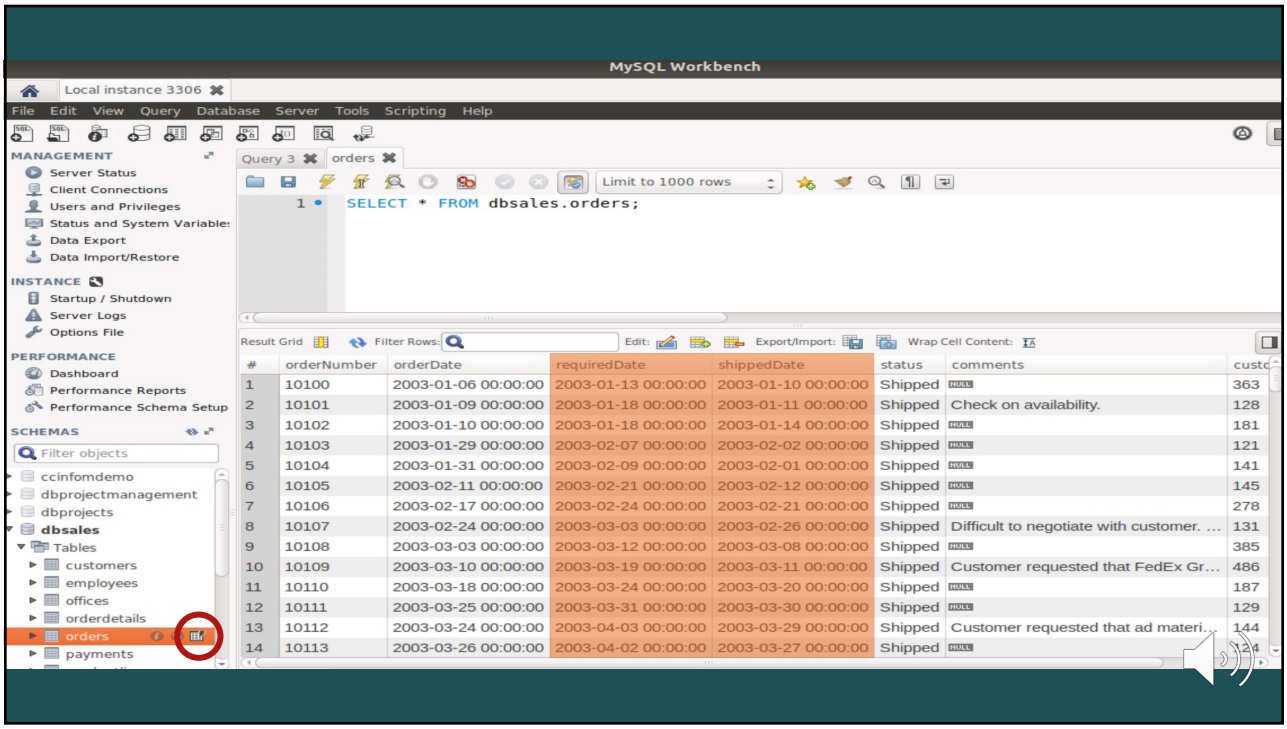
2. Get the list of orders which were not fulfilled on time.

29

dbsales schema



30



31

SQL Practice Exercises

2. Get the list of orders which were not fulfilled on time.

```
SELECT      *
FROM        orders
WHERE       requireddate < shippeddate;
```


32

SQL Practice Exercises

2. Get the list of orders which were not fulfilled on time.

```
SELECT      *
FROM        orders
WHERE       shippeddate > requireddate;
```


33

#	orderNumber	orderDate	requiredDate	shippedDate	status	comments
1	10165	2003-10-22 00:00:00	2003-10-31 00:00:00	2003-12-26 00:00:00	Shipped	This order was on hold because cu…
2	10427	2016-09-15 16:01:36	2013-01-01 00:00:00	2016-09-27 00:00:00	In Process	Test Order

Returns 2 Rows

34

SQL Practice Exercises

2. Get the list of orders which were not fulfilled on time.

```
SELECT      *
FROM        orders
WHERE       requireddate < shippeddate
ORDER BY    ordernumber;
```

Information Technology Department
De La Salle University

35

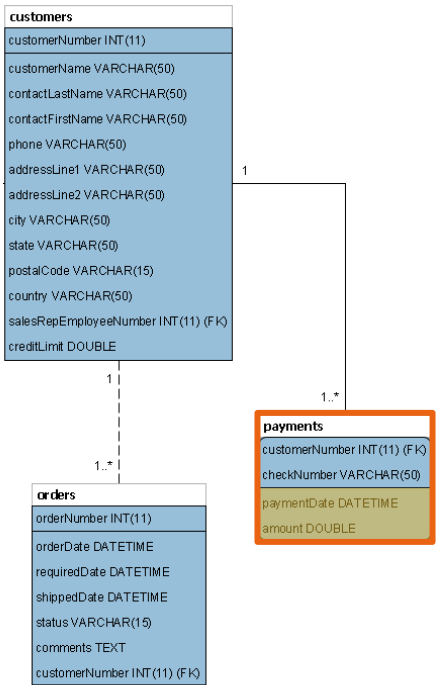
SQL Practice Exercises

3. Get the total amount of payments made in 2003.

Information Technology Department
De La Salle University

36

dbsales schema



37

SQL Practice Exercises

3. Get the total amount of payments made in 2003.

```
SELECT      SUM(amount)
FROM        payments
WHERE       YEAR(paymentdate) = 2003;
```


38

SQL Practice Exercises

3. Get the total amount of payments made in 2003.

```
SELECT      SUM(amount)
FROM        payments
WHERE       YEAR(paymentdate) = 2003;
```


39

SQL Practice Exercises

3. Get the total amount of payments made in 2003.

```
SELECT      SUM(amount)
FROM        payments
WHERE       YEAR(paymentdate) = 2003;
```

YEAR(date)

40

#	SUM(amount)
1	3250217.7000000007

Returns 1 Row

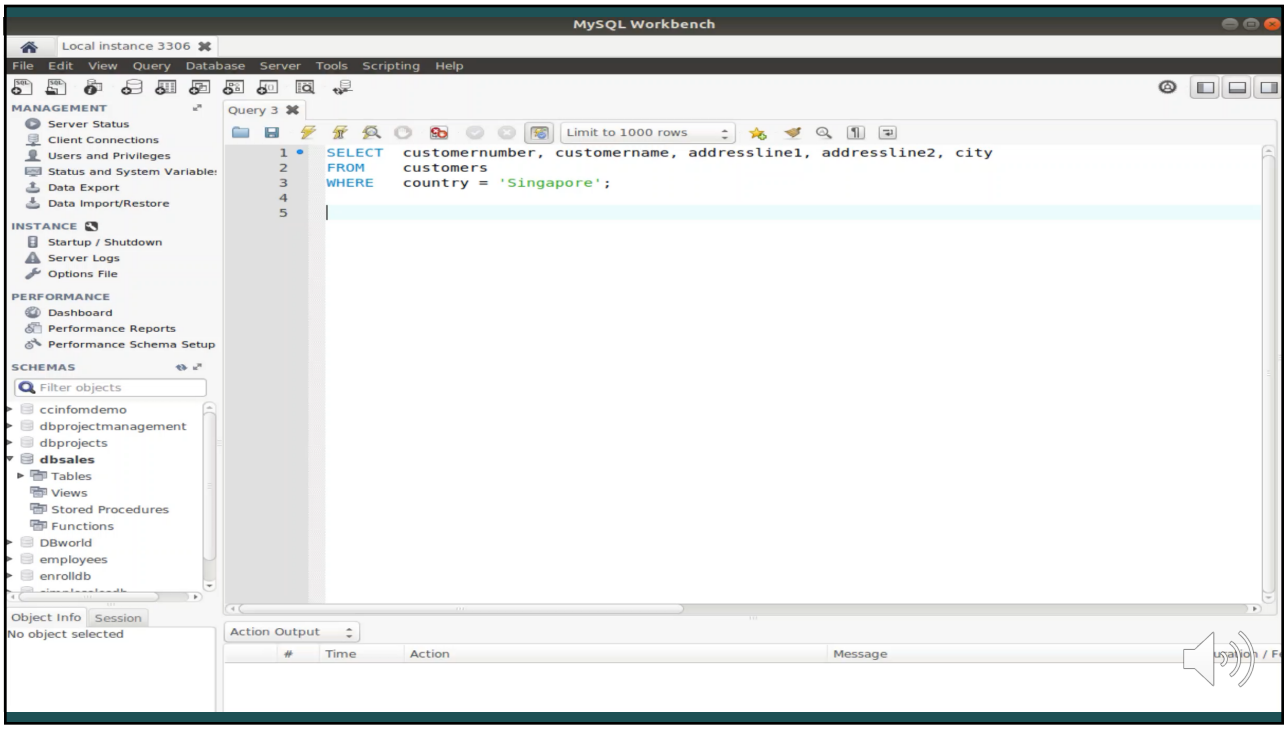
41

SQL Practice Exercises

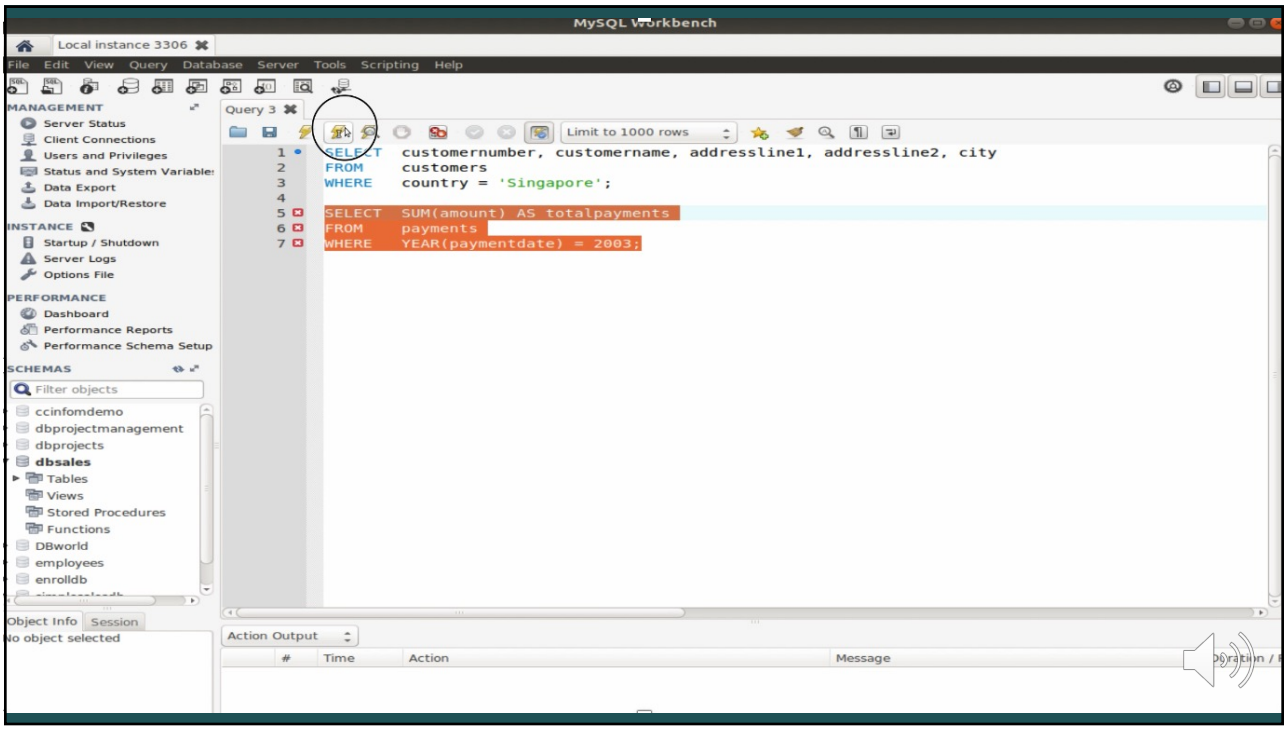
3. Get the total amount of payments made in 2003.

```
SELECT      SUM(amount) AS totalpayments
FROM        payments
WHERE       YEAR(paymentdate) = 2003;
```

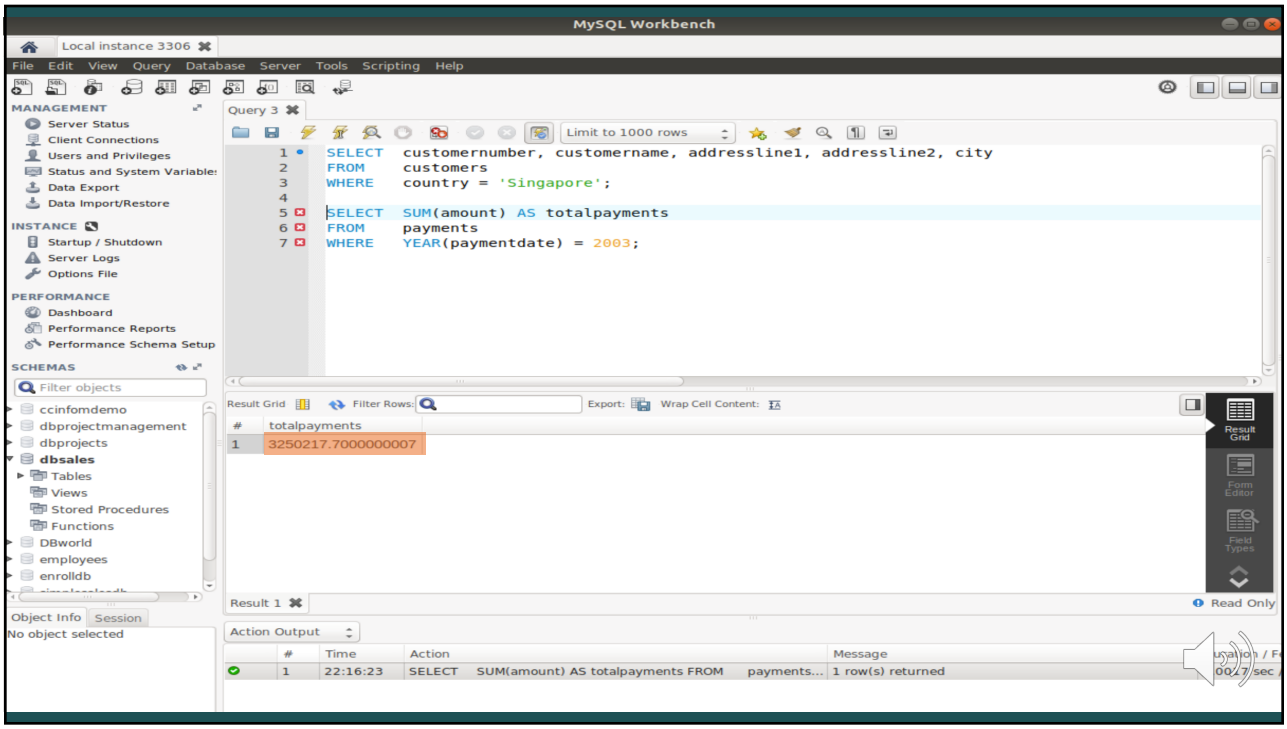
42



43


MySQL Workbench
Local instance 3306
File Edit View Query Database Server Tools Scripting Help
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
ccinfomdemo
dbprojectmanagement
dbprojects
dbsales
Tables
Views
Stored Procedures
Functions
DBworld
employees
enrolldb
Object Info
Session
No object selected
Query 3
Limit to 1000 rows
1 SELECT customernumber, customername, addressline1, addressline2, city
2 FROM customers
3 WHERE country = 'Singapore';
4
5 SELECT SUM(amount) AS totalpayments
6 FROM payments
7 WHERE YEAR(paymentdate) = 2003;
Action Output
Time Action Message

44


MySQL Workbench
Local instance 3306
File Edit View Query Database Server Tools Scripting Help
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
ccinfomdemo
dbprojectmanagement
dbprojects
dbsales
Tables
Views
Stored Procedures
Functions
DBworld
employees
enrolldb
Object Info
Session
No object selected
Query 3
Limit to 1000 rows
1 SELECT customernumber, customername, addressline1, addressline2, city
2 FROM customers
3 WHERE country = 'Singapore';
4
5 SELECT SUM(amount) AS totalpayments
6 FROM payments
7 WHERE YEAR(paymentdate) = 2003;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
totalpayments
1 3250217.7000000007
Result 1
Read Only
Action Output
Time Action Message
1 22:16:23 SELECT SUM(amount) AS totalpayments FROM payments... 1 row(s) returned

45

SQL Practice Exercises

3. Get the total amount of payments made in 2003.

```
SELECT      FORMAT(SUM(amount),2) AS totalpayments
FROM        payments
WHERE       YEAR(paymentdate) = 2003;
```

FORMAT(number, decimal places)

46

#	totalpayments
1	3,250,217.70

Returns 1 Row

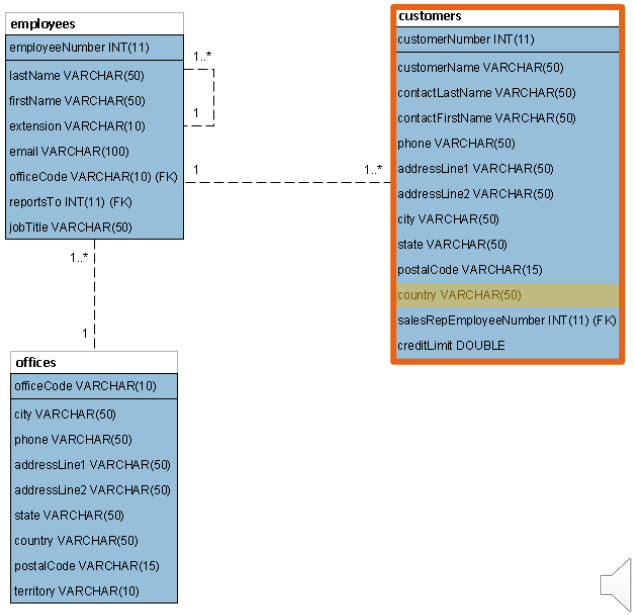
47

SQL Practice Exercises

4. Get the total number of customers per country.

48

dbsales schema



49

SQL Practice Exercises

4. Get the total number of customers per country.

```
SELECT      country, COUNT(customernumber) AS numofcustomers
FROM        customers
GROUP BY    country;
```


50

SQL Practice Exercises

4. Get the total number of customers per country.

```
SELECT      country, COUNT(customernumber) AS numofcustomers
FROM        customers
GROUP BY    country;
```


51

SQL Practice Exercises

4. Get the total number of customers per country.

```
SELECT      country, COUNT(customernumber) AS numofcustomers
FROM        customers
GROUP BY    country;
```


52

SQL Practice Exercises

4. Get the total number of customers per country.

```
SELECT      country, COUNT(customernumber) AS numofcustomers
FROM        customers
GROUP BY    country;
```

Always pair a summative function with GROUP BY.

GROUP BY should match the SELECT clause.

53

#	country	numofcustomers
1	Australia	5
2	Austria	2
3	Belgium	2
4	Canada	3
5	Denmark	2
6	Finland	3
7	France	12
8	Germany	13
9	Hong Kong	1
10	Ireland	2
11	Israel	1
12	Italy	4
13	Japan	2
14	Netherlands	1
15	New Zealand	4

Returns 27 Rows

54

SQL Practice Exercises

4. Get the total number of customers per country.

```
SELECT      country, COUNT(customernumber) AS numofcustomers
FROM        customers
GROUP BY    country
ORDER BY    country;
```

Information Technology Department
De La Salle University

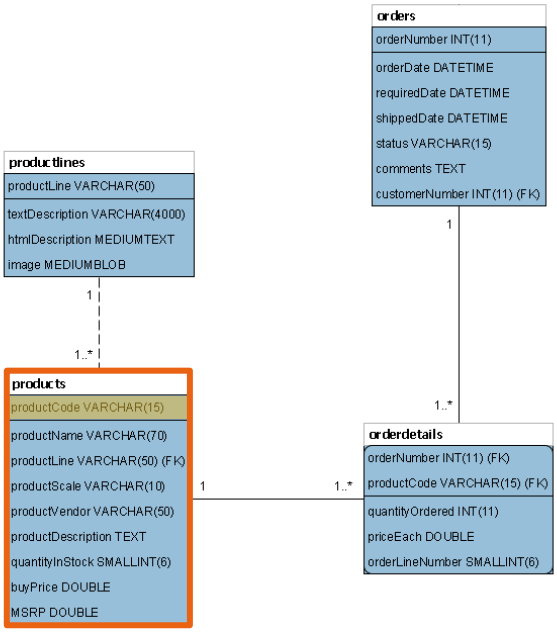
55

SQL Practice Exercises

5. Get the number of products per vendor.

56

dbsales schema



57

SQL Practice Exercises

5. Get the number of products per vendor.

```
SELECT      productvendor, COUNT(productcode) AS numofproducts
FROM        products
GROUP BY    productvendor
ORDER BY    productvendor;
```


58

SQL Practice Exercises

5. Get the number of products per vendor.

```
SELECT      productvendor, COUNT(productcode) AS numofproducts
FROM        products
GROUP BY    productvendor
ORDER BY    productvendor;
```


59

SQL Practice Exercises

5. Get the number of products per vendor.

```
SELECT      productvendor, COUNT(productcode) AS numofproducts
FROM        products
GROUP BY    productvendor
ORDER BY    productvendor;
```

60

SQL Practice Exercises

5. Get the number of products per vendor.

```
SELECT      productvendor, COUNT(productcode) AS numofproducts
FROM        products
GROUP BY    productvendor
ORDER BY    productvendor;
```

61

#	productvendor	numofproducts
1	Autoart Studio Design	8
2	Carousel DieCast Legends	9
3	Classic Metal Creations	10
4	Exoto Designs	9
5	Gearbox Collectibles	9
6	Highway 66 Mini Classics	9
7	Min Lin Diecast	8
8	Motor City Art Classics	9
9	Red Start Diecast	7
10	Second Gear Diecast	8
11	Studio M Art Models	8
12	Unimax Art Galleries	8
13	Welly Diecast Productions	8

Returns 13 Rows

62

Questions?

PLEASE FEEL FREE TO USE YOUR MICROPHONE.

Information Technology Department
De La Salle University

63

End of Lesson Exercise

1. Generate a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.

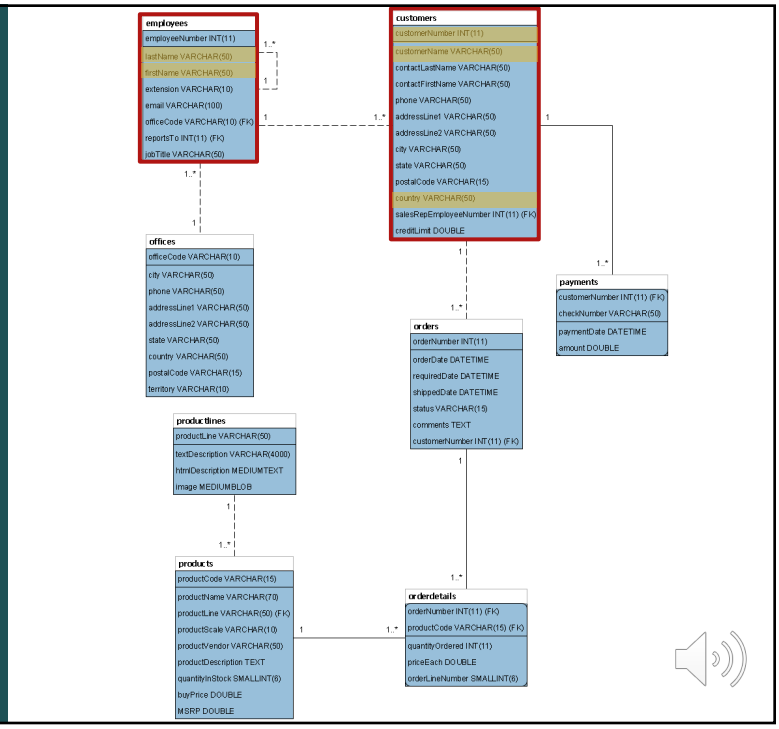
64

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

65

dbsales schema



66

End of Lesson Exercise

1. Generate a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.

```
FROM      customers c
          employees e
```


67

End of Lesson Exercise

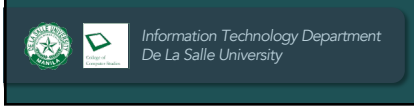
1. Generate a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.

```
FROM      customers c
JOIN      employees e ON c.salesRepEmployeeNumber =
          e.employeeNumber
```

68

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on



No other conditions to be satisfied.

69

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

70

End of Lesson Exercise

1. Generate a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.

```
SELECT   c.customerNumber, c.customerName, c.country,
         e.firstName, e.lastName
FROM     customers c
JOIN     employees e ON c.salesRepEmployeeNumber =
         e.employeeNumber
```


71

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

Information Technology Department
De La Salle University

72

End of Lesson Exercise

1. Generate a report showing the customers (customer number and complete name), the country they are located and the complete name of the sales representative handling her/him.

```
SELECT    c.customerNumber, c.customerName, c.country,
          e.firstName, e.lastName
FROM      customers c
JOIN      employees e ON c.salesRepEmployeeNumber =
          e.employeeNumber
ORDER BY c.customerNumber;
```

Information Technology Department
De La Salle University

73

customerNu...	customerName	country	firstName	lastName
103	Atelier graphique	France	Gerard	Hernandez
112	Signal Gift Stores	USA	Leslie	Thompson
114	Australian Collectors, Co.	Australia	Andy	Fixter
119	La Rochelle Gifts	France	Gerard	Hernandez
121	Baane Mini Imports	Norway	Barry	Jones
124	Mini Gifts Distributors Ltd.	USA	Leslie	Jennings
128	Blauer See Auto, Co.	Germany	Barry	Jones
129	Mini Wheels Co.	USA	Leslie	Jennings
131	Land of Toys Inc.	USA	George	Vanauf
141	Euro+ Shopping Channel	Spain	Gerard	Hernandez
144	Volvo Model Replicas, Co	Sweden	Barry	Jones
145	Danish Wholesale Imports	Denmark	Pamela	Castillo
146	Saveley & Henriot, Co.	France	Loui	Bondur
148	Dragon Souveniers, Ltd.	Singapore	Mami	Nishi
151	Muscle Machine Inc	USA	Foon Yue	Tseng
157	Diecast Classics Inc.	USA	Steve	Patterson
161	Technics Stores Inc.	USA	Leslie	Jennings
166	Handji Gifts& Co	Singapore	Peter	Marsh
167	Herkku Gifts	Norway	Barry	Jones

Returns 100 Rows

74

End of Lesson Exercise

CHANGE ROW LIMIT

Information Technology Department
De La Salle University

75

End of Lesson Exercise

NO DATABASE SELECTED

Message
Error Code: 1046. Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.

76

End of Lesson Exercise

WRONG DATABASE SELECTED/WRONG TABLE NAME SPECIFIED

Response
Error Code: 1146. Table 'ccinfomdemo.customers' doesn't exist

77

Questions?

PLEASE FEEL FREE TO USE YOUR MICROPHONE.

Information Technology Department
De La Salle University

78

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

Information Technology Department
De La Salle University

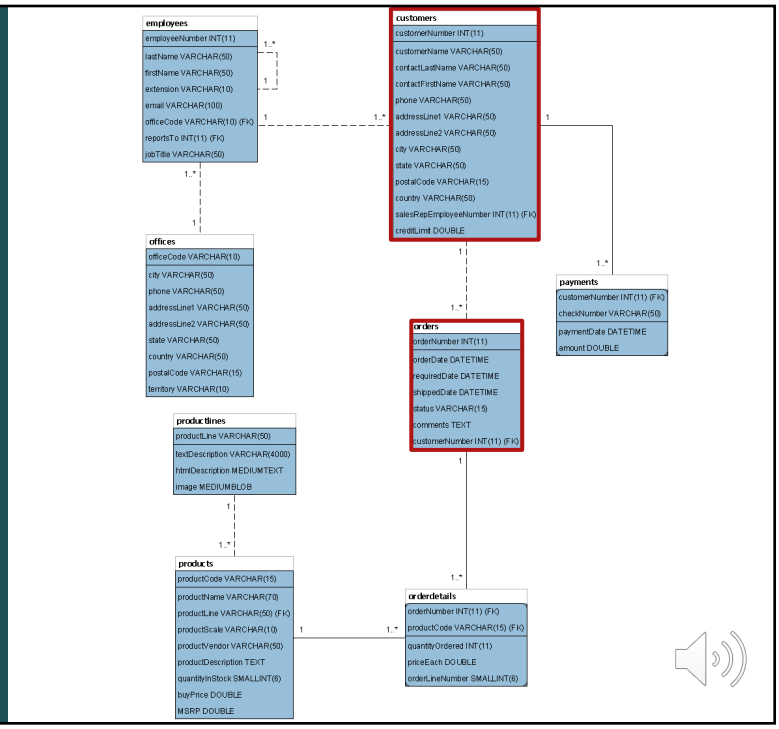
79

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

80

dbsales schema



81

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
FROM        orders o
            customers c
```

82

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
FROM        orders o
JOIN        customers c ON o.customerNumber = c.customerNumber
```

83

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on


Information Technology Department
De La Salle University

84

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.


Information Technology Department
De La Salle University

85

End of Lesson Exercise

- View the contents of the ORDERS table to check all the possible order statuses

```
SELECT *
FROM   orders;
```

- Order Status: Cancelled, Disputed, On Hold, Resolved, Shipped
- Order is Completed = Order Status is 'Shipped'

86

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
FROM        orders o
JOIN        customers c ON o.customerNumber = c.customerNumber
WHERE       o.status = 'Shipped'
```


87

End of Lesson Exercise

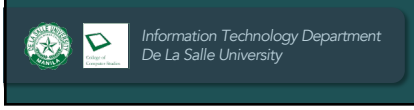
2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
FROM        orders o
JOIN        customers c ON o.customerNumber = c.customerNumber
WHERE       o.status = 'Shipped' AND o.shippedDate BETWEEN '2005-05-01' AND '2005-05-31'
```

88

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on



89

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
SELECT      o.orderNumber, o.orderDate, o.shippedDate, o.status, c.customerName
FROM        orders o
JOIN        customers c ON o.customerNumber = c.customerNumber
WHERE       o.status = 'Shipped' AND o.shippedDate BETWEEN '2005-05-01' AND '2005-05-31'
```


90

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on


91

End of Lesson Exercise

2. Generate the list of orders completed in May 2005 showing the order number, the order date, shipped date, and the complete name of the customer who made the order.

```
SELECT      o.orderNumber, o.orderDate, o.shippedDate, c.customerName
FROM        orders o
JOIN        customers c ON o.customerNumber = c.customerNumber
WHERE       o.status = 'Shipped' AND o.shippedDate BETWEEN '2005-05-01' AND '2005-05-31'
ORDER BY    o.orderNumber;
```

Information Technology Department
De La Salle University

92

orderNumber	orderDate	shippedDate	customerName
10411	2005-05-01 00:00:00	2005-05-06 00:00:00	Québec Home Shopping Network
10412	2005-05-03 00:00:00	2005-05-05 00:00:00	Euro+ Shopping Channel
10413	2005-05-05 00:00:00	2005-05-09 00:00:00	Gift Depot Inc.
10416	2005-05-10 00:00:00	2005-05-14 00:00:00	L'ordine Souveniers
10418	2005-05-16 00:00:00	2005-05-20 00:00:00	Extreme Desk Decorations, Ltd
10419	2005-05-17 00:00:00	2005-05-19 00:00:00	Salzburg Collectables

Returns 6 Rows

93

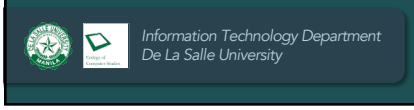
End of Lesson Exercise

3. Generate the list of products below 300 pieces. Show in the list the product's code and name, and the product line the product belongs to.

95

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on



Please type your answer in the Chat section.

96

End of Lesson Exercise

3. Generate the list of products below 300 pieces. Show in the list the product's code and name, and the product line the product belongs to.

```
SELECT   p.productCode, p.productName, p.productLine,
         p.quantityInStock
FROM     products p
WHERE    p.quantityInStock < 300
ORDER BY p.productCode;
```

101

productCode	productName	productLine	quantityInStock
S12_1099	1968 Ford Mustang	Classic Cars	68
S24_2000	1960 BSA Gold Star DBD34	Motorcycles	15
S32_1374	1997 BMW F650 ST	Motorcycles	178
S32_4289	1928 Ford Phaeton Deluxe	Vintage Cars	136

Returns 4 Rows

102

End of Lesson Exercise

4. Generate the complete directory of employees, showing their complete name, email, extension number, the office code they belong to as well as the office phone number.

104

End of Lesson Exercise

4. Generate the complete directory of employees, showing their complete name, email, extension number, the office code they belong to as well as the office phone number.

```
SELECT   e.lastName, e.firstName, e.email, e.extension,
         e.officeCode, o.phone
FROM     employees e
JOIN     offices o ON e.officeCode=o.officeCode
ORDER BY e.lastName, e.firstName;
```


110

lastName	firstName	email	extension	officeCode	phone
Bondur	Gerard	gbondur@classicmodelcars.com	x5408	4	+33 14 723 4404
Bondur	Loui	lbondur@classicmodelcars.com	x6493	4	+33 14 723 4404
Bott	Larry	lbott@classicmodelcars.com	x2311	7	+44 20 7877 2041
Bow	Anthony	abow@classicmodelcars.com	x5428	1	+1 650 219 4782
Castillo	Pamela	pcastillo@classicmodelcars.com	x2759	4	+33 14 723 4404
Firrelli	Jeff	jfirrelli@classicmodelcars.com	x9273	1	+1 650 219 4782
Firrelli	Julie	jfirrelli@classicmodelcars.com	x2173	2	+1 215 837 0825
Fixter	Andy	afixter@classicmodelcars.com	x101	6	+61 2 9264 2451
Gerard	Martin	mgerard@classicmodelcars.com	x2312	4	+33 14 723 4404
Hernandez	Gerard	ghernande@classicmodelcars.com	x2028	4	+33 14 723 4404
Jennings	Leslie	ljennings@classicmodelcars.com	x3291	1	+1 650 219 4782
Jones	Barry	bjones@classicmodelcars.com	x102	7	+44 20 7877 2041
Kato	Yoshimi	ykato@classicmodelcars.com	x102	5	+81 33 224 5000
King	Tom	tking@classicmodelcars.com	x103	6	+61 2 9264 2451
Marsh	Peter	pmarsh@classicmodelcars.com	x102	6	+61 2 9264 2451
Murphy	Diane	dmurphy@classicmodelcars.com	x5800	1	+1 650 219 4782
Nishi	Mami	mnishi@classicmodelcars.com	x101	5	+81 33 224 5000
Patterson	Mary	mpatterso@classicmodelcars.com	x4611	1	+1 650 219 4782
Patterson	Steve	spatterson@classicmodelcars.com	x4334	2	+1 215 837 0825
Patterson	William	wpatterson@classicmodelcars.com	x4871	6	+61 2 9264 2451
Thompson	Leslie	lthompson@classicmodelcars.com	x4065	1	+1 650 219 4782
Tseng	Foon Yue	ftseng@classicmodelcars.com	x2248	3	+1 212 555 3000
Vanauf	George	gvanauf@classicmodelcars.com	x4102	3	+1 212 555 3000

Returns 23 Rows

111

Introduction to Outer Joins

Information Technology Department
De La Salle University

115

What is an Outer Join?

- Taking all the records of one table regardless whether they have a matching record/s on another table.

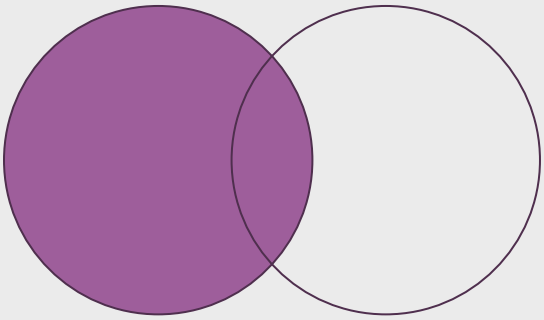
LEFT JOIN | RIGHT JOIN

Information Technology Department
De La Salle University

116

LEFT JOIN

```
SELECT      columns
FROM        table1
LEFT JOIN   table2
            ON column1=column2;
```

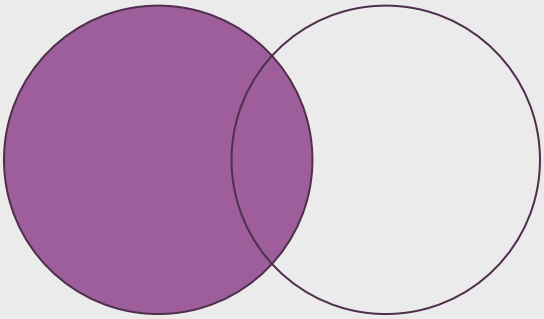


Information Technology Department
De La Salle University

117

LEFT JOIN

```
SELECT      columns
FROM        table1
LEFT JOIN   table2
            ON column1=column2;
```

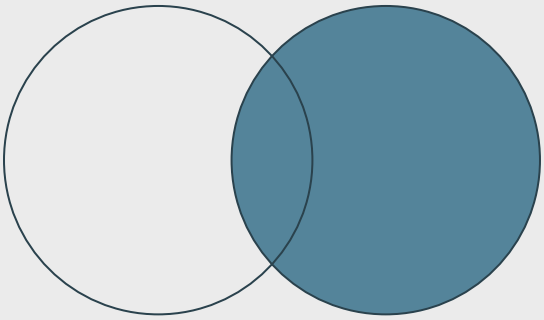


Information Technology Department
De La Salle University

118

RIGHT JOIN

```
SELECT      columns
FROM        table1
RIGHT JOIN  table2
            ON column1=column2;
```

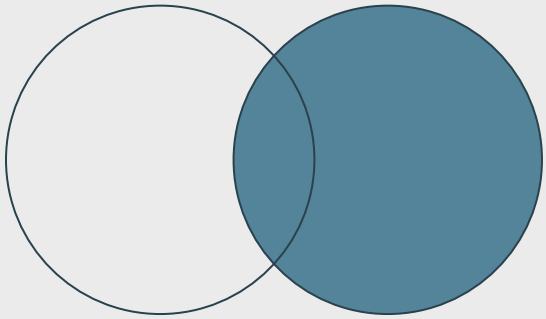


Information Technology Department
De La Salle University

119

RIGHT JOIN

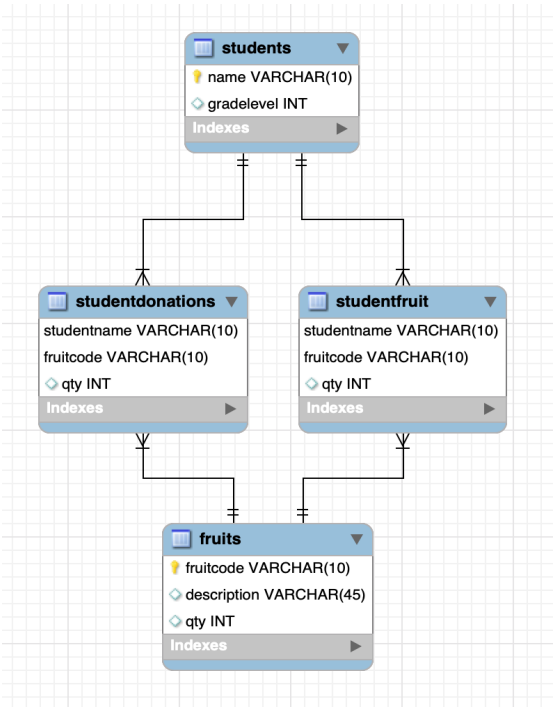
```
SELECT      columns
FROM        table1
RIGHT JOIN  table2
            ON column1=column2;
```



Information Technology Department
De La Salle University

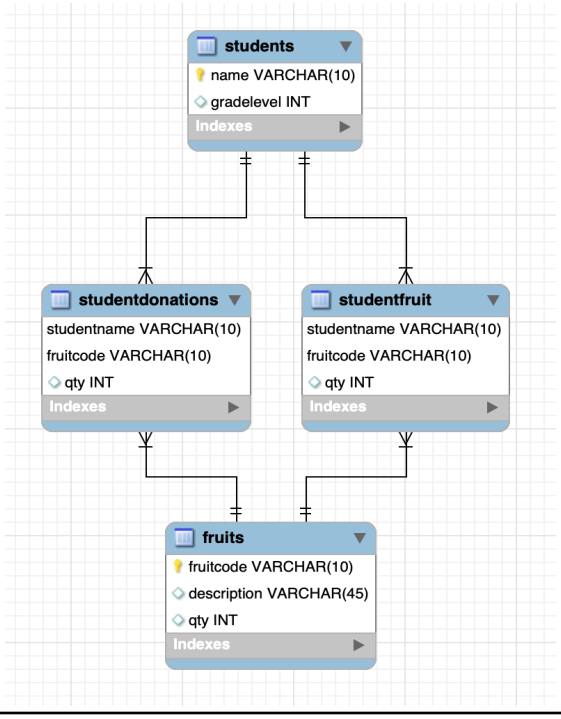
120

dbsimple schema



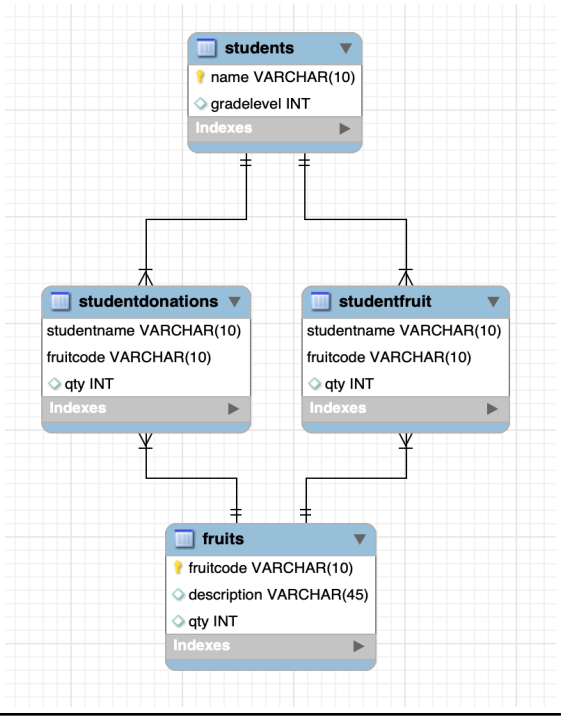
121

dbsimple schema



122

dbsimple schema



123

Fruits

fruitcode	description	qty
APPLE	Red Fruit	20
BANANA	Yellow Fruit	10
MANGO	Yelow Fruit	10
TOMATO	Red Fruit	20

Students

name	gradelevel
ALEXANDRA	2
ANGELA	6
CHRISTIAN	4
GERONE	2
JEHOVAH	5
LEON	5
LISSA	4
MARGARETH	3

StudentDonations

studentname	fruitcode	qty
ANGELA	MANGO	4
GERONE	BANANA	10
LEON	APPLE	4
LISSA	APPLE	4

StudentFruits

studentname	fruitcode	qty
ANGELA	MANGO	3
JEHOVAH	APPLE	2
LEON	APPLE	3

124

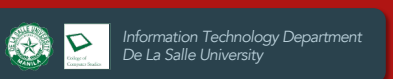
Try This

List all the students, and the donations (if applicable) they gave.

125

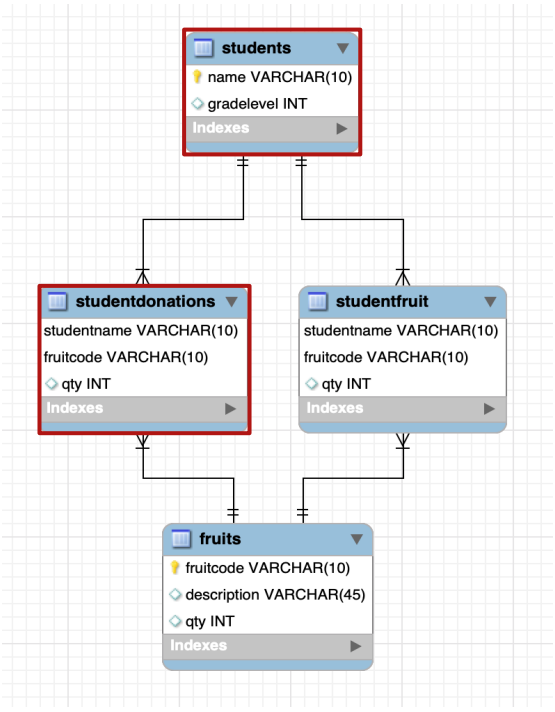
Try This

List all the students, and the donations (if applicable) they gave.



126

dbsimple schema

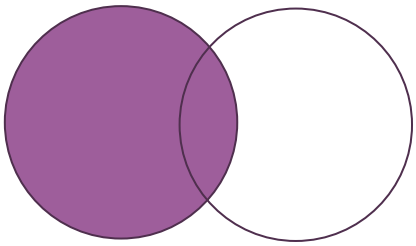


127

Practice Exercise

List all the students, and the donations (if applicable) they gave.

LEFT JOIN

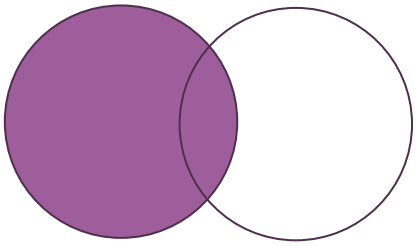


`table1` `table2`

128

Practice Exercise

List all the students, and the donations (if applicable) they gave.



`table1:` `table2`
`students`

129

Practice Exercise

List all the students, and the donations (if applicable) they gave.

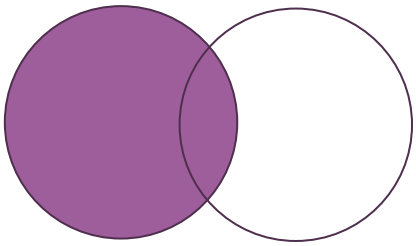


table1: students

table2: studentdonations

130

Practice Exercise

List all the students, and the donations (if applicable) they gave.

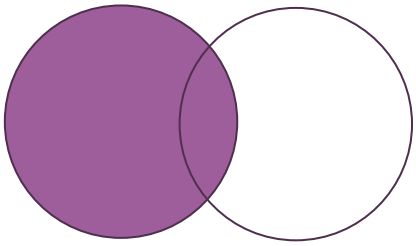


table1: students

table2: studentdonations

Information Technology Department
De La Salle University

131

Fruits

fruitcode	description	qty
APPLE	Red Fruit	20
BANANA	Yellow Fruit	10
MANGO	Yelow Fruit	10
TOMATO	Red Fruit	20

Students

name	gradelevel
ALEXANDRA	2
ANGELA	6
CHRISTIAN	4
GERONE	2
JEHOVAH	5
LEON	5
LISSA	4
MARGARETH	3

StudentDonations

studentname	fruitcode	qty
ANGELA	MANGO	4
GERONE	BANANA	10
LEON	APPLE	4
LISSA	APPLE	4

StudentFruits

studentname	fruitcode	qty
ANGELA	MANGO	3
JEHOVAH	APPLE	2
LEON	APPLE	3

Did all students make a donation?

132

Fruits

fruitcode	description	qty
APPLE	Red Fruit	20
BANANA	Yellow Fruit	10
MANGO	Yelow Fruit	10
TOMATO	Red Fruit	20

Students

name	gradelevel
ALEXANDRA	2
ANGELA	6
CHRISTIAN	4
GERONE	2
JEHOVAH	5
LEON	5
LISSA	4
MARGARETH	3

StudentDonations

studentname	fruitcode	qty
ANGELA	MANGO	4
GERONE	BANANA	10
LEON	APPLE	4
LISSA	APPLE	4

StudentFruits

studentname	fruitcode	qty
ANGELA	MANGO	3
JEHOVAH	APPLE	2
LEON	APPLE	3

Did all students make a donation?

133

Practice Exercise

List all the students, and the donations (if applicable) they gave.

```
FROM        students s
LEFT JOIN   studentdonations sd
            ON s.name=sd.studentname
```


134

Practice Exercise

List all the students, and the donations (if applicable) they gave.

```
SELECT      s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM        students s
LEFT JOIN   studentdonations sd
            ON s.name=sd.studentname
```


135

Practice Exercise

List all the students, and the donations (if applicable) they gave.

```
SELECT      s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM        students s
LEFT JOIN   studentdonations sd
            ON s.name=sd.studentname
ORDER BY    s.name;
```


Practice Exercise

List all the students, and the donations (if applicable) they gave.

```
SELECT      s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM        students s
LEFT JOIN   studentdonations sd
            ON s.name=sd.studentname
ORDER BY    s.name;
```


name	gradelevel	fruitcode	qty
ALEXANDRA	2	NULL	NULL
ANGELA	6	MANGO	4
CHRISTIAN	4	NULL	NULL
GERONE	2	BANANA	10
JEHOVAH	5	NULL	NULL
LEON	5	APPLE	4
LISSA	4	APPLE	4
MARGARETH	3	NULL	NULL

Returns 8 Rows

138

About Outer Joins

LEFT JOINs can be written as RIGHT JOINs and vice versa.

Information Technology Department
De La Salle University

139

Practice Exercise

List all the students, and the donations (if applicable) they gave.

RIGHT JOIN

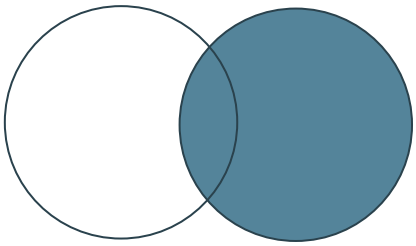


table1 table2

140

Practice Exercise

List all the students, and the donations (if applicable) they gave.

RIGHT JOIN

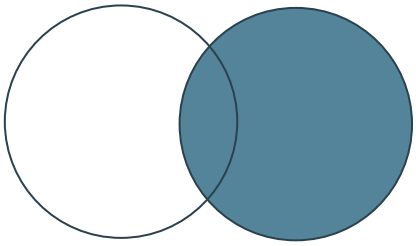


table2:
students

table1:
studentdonations

141

Practice Exercise

List all the students, and the donations (if applicable) they gave.

```
SELECT      s.name, s.gradelevel, sd.fruitcode, sd.qty
FROM        studentdonations sd
RIGHT JOIN  students s
            ON s.name=sd.studentname
ORDER BY    s.name;
```

142

name	gradelevel	fruitcode	qty
ALEXANDRA	2	NULL	NULL
ANGELA	6	MANGO	4
CHRISTIAN	4	NULL	NULL
GERONE	2	BANANA	10
JEHOVAH	5	NULL	NULL
LEON	5	APPLE	4
LISSA	4	APPLE	4
MARGARETH	3	NULL	NULL

Returns 8 Rows

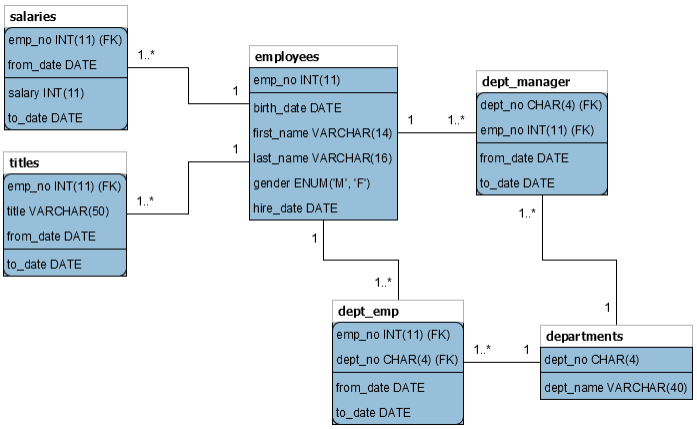
143

Practice Exercise

1. Create a new database using the script file "dbemployees.sql".
2. Using the employees database/schema, write the SQL statement that will satisfy each data requirement in the succeeding slide.

144

employees schema



145

Practice Exercise

1. List all the employees, the department currently being or previously managed, and the date of assignment as manager (if they manage a department). Sort the list by the date of assignment as manager (latest assignment first). (Returns 300,326 rows)
2. List all the employees (employee number and complete name), and the titles they were assigned (if they were assigned titles). Sort the list such that employees without titles appear first and should they have titles, sort it by employee name). (Returns 443,310 rows)
3. List all the departments (department name), and all the current and previous managers (employee number and date of assignment). Sort the list by department name and latest assignment first. (Returns 24 rows)

148

149

Practice Exercise

1. The sales manager wants to request for products whose quantity is below 100. Provide the information the manager needs by generating a list of products (product code, product line, product name, and vendor) that are below 100 pieces in stock. The product line and the product name should be combined in a single column with the following format: "product line: product name". (Returns 2 rows)
   NOTE: Use the CONCAT() function.
   `CONCAT(string1,string2,…)`
2. The sales manager wants to see all products which are 1999 models. (Returns 2 rows)
   NOTE: Use the SUBSTRING()/SUBSTR() function.
   `SUBSTR(string,start,length)`
3. The sales manager wants to see all products which are 1940-1950 models. (Returns 8 rows)
   NOTE: Use the SUBSTRING()/SUBSTR() function.
   `SUBSTR(string,start,length)`

Information Technology Department
De La Salle University

150

151

Questions?

Information Technology Department
De La Salle University

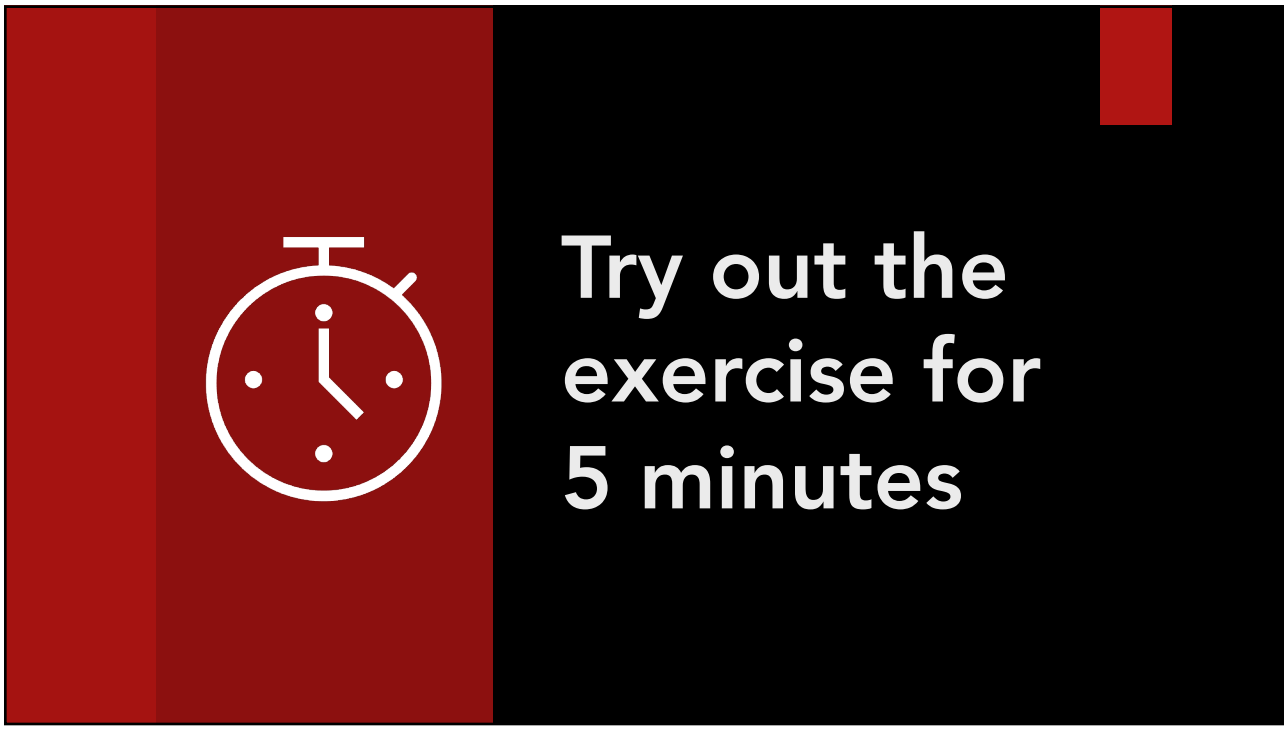
152

Practice Exercise

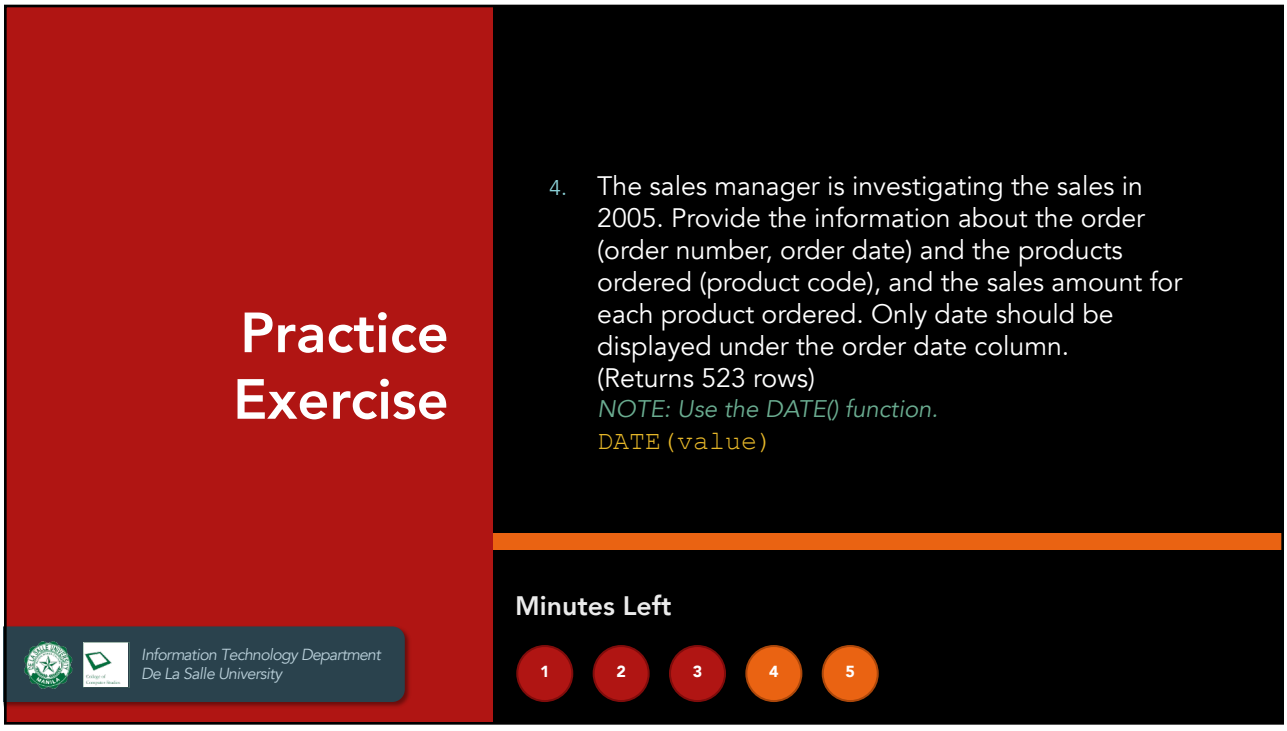
4. The sales manager is investigating the sales in 2005. Provide the information about the order (order number, order date) and the products ordered (product code), and the sales amount for each product ordered. Only date should be displayed under the order date column. (Returns 523 rows)
   NOTE: Use the DATE() function.
   `DATE(value)`

Information Technology Department
De La Salle University

153


Try out the exercise for 5 minutes

154


Practice Exercise
4. The sales manager is investigating the sales in 2005. Provide the information about the order (order number, order date) and the products ordered (product code), and the sales amount for each product ordered. Only date should be displayed under the order date column. (Returns 523 rows)
NOTE: Use the DATE() function.
DATE(value)
Minutes Left
1 2 3 4 5
Information Technology Department
De La Salle University

155

156

Questions?

Information Technology Department
De La Salle University

157

Thinking Discipline in Writing SQL Statements

Information Technology Department
De La Salle University

158

Pre-SQL Writing Tasks

1. Review the data model
2. Review the data inside the database (especially dates, status, codes, ID, monetary values)
   - DLSU ID is not just a number, it contains the year of entry and the student sequence (e.g. 119XXXXX means entered 2019 and XXXXX is the student sequence)
   - Some status are coded (e.g. 1 for Ordered, 2 for Prepared, 3 for Shipped, 4 for Delivered)
   - Some dates are designed not to be empty and empty may be replaced by a sentinel value (e.g. to_date is 9999-01-01 if it means empty)

Information Technology Department
De La Salle University

159

Sequence of Clauses in an SQL SELECT Statement

CLAUSE	DESCRIPTION
SELECT	What data will appear in the information result
FROM	Where will the data be coming from
WHERE	What conditions will be used to filter the data
ORDER BY	What fields will the information be sorted on

160

Recommended Sequence of Thinking

CLAUSE	DESCRIPTION
1. FROM	Identify where will the data you need come from
2. WHERE	Identify what conditions you need to filter the data
3. SELECT	Identify the fields needed to represent the information requirement
4. ORDER BY	Identify what fields in the SELECT clause, you want the information to be sorted on

161

Other Considerations

1. What will the result look like?
2. What method do you need to use to retrieve data from tables?
   - How many tables to do you need to use?
   - Do you need to combine them?
     - Do you need to use INNER JOIN (Join)?
     - Do you need to use OUTER JOIN (Left Join)?
3. Do you need to simply refer to the data that other tables may have?*
4. Does the requirement need the results to be grouped in some way and need summative information?

162

Common Mistakes in Writing SQL Statements

163

End of Lesson Exercise

NO DATABASE SELECTED

Message
Error Code: 1046. Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.

164

End of Lesson Exercise

WRONG DATABASE SELECTED/WRONG TABLE NAME SPECIFIED

Response
Error Code: 1146. Table 'ccinfomdemo.customers' doesn't exist

165

End of Lesson Exercise

USING A SUMMATIVE FUNCTION WITHOUT GROUP BY

Response
Error Code: 1140. In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'employees.d.dept_name';

Information Technology Department
De La Salle University

166

Answer the SQL Joins Exercise in AnimoSpace

dbemployees.sql

Information Technology Department
De La Salle University

167

References

- Connolly, T. & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Ed.* Harlow, Essex: Addison-Wesley [QA 76.9VD26 C66 2015]
- MySQL Workbench Manual *https://dev.mysql.com/doc/workbench/en/wb-sql-editor.html*

168

Questions?

PLEASE POST YOUR QUESTIONS IN THE DISCUSSION BOARD CREATED FOR THIS TOPIC.

169


CCINFOM
Thank you & keep safe.
Information Technology Department
College of Computer Studies
De La Salle University

170