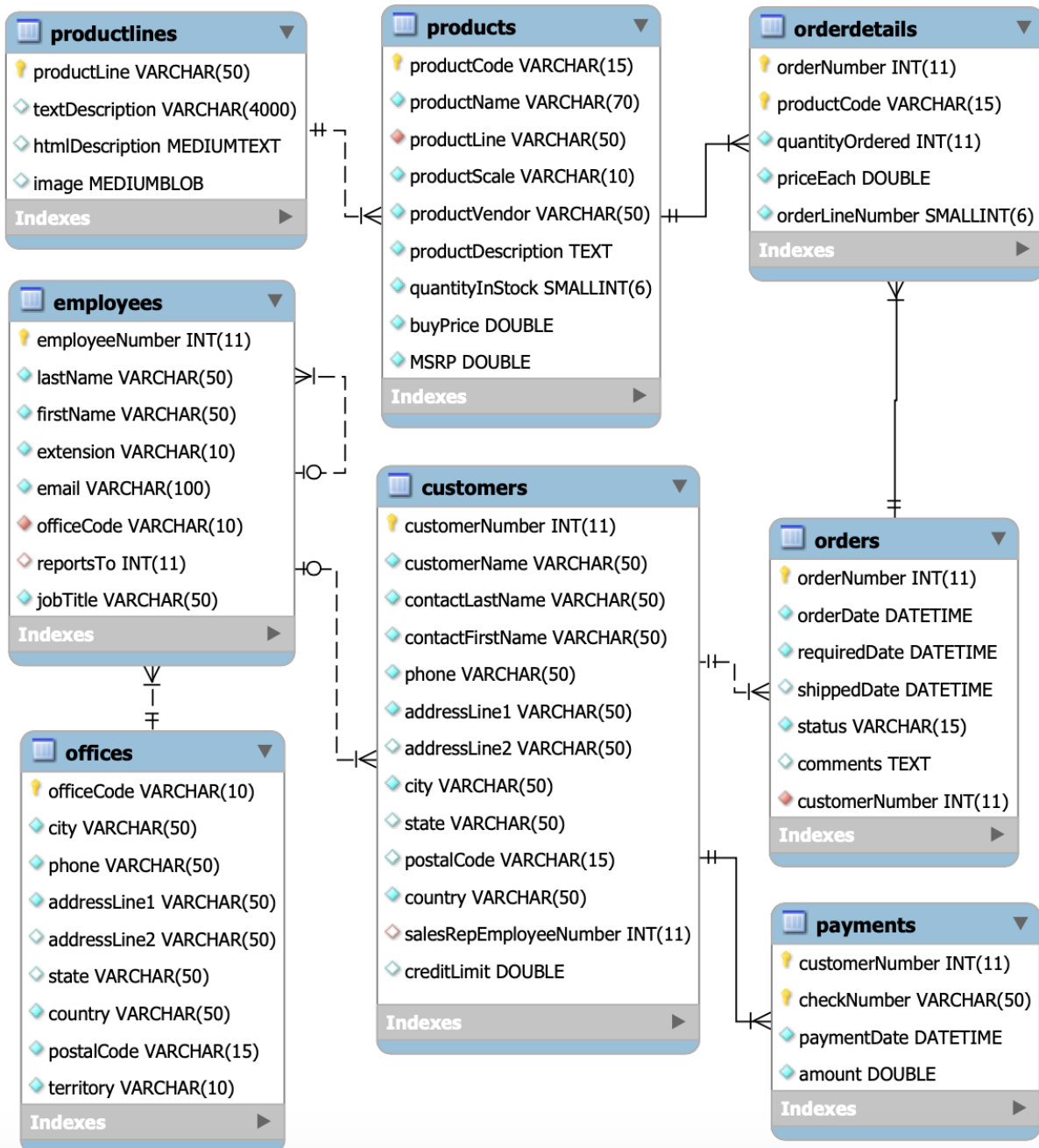# LAB 03

BASIC QUERIES (PART 1)

❖ **Main contents**

- SELECT statement: syntax and usage
- WHERE clause
- Removing duplicate rows with DISTINCT statement
- Limiting query results by using LIMIT statement

1. Install the sample database

The *classicmodels* database is a retailer of scale models of classic cars database. It contains typical business data such as customers, products, sales orders, sales order line items, etc. The sample database includes the following tables:

- *Customers*: Store information about customers.
- *Products*: Store a list of scale model cars.
- *ProductLines*: Store a list of product line categories.
- *Orders*: Store sales orders placed by customers.
- *OrderDetails*: Store detailed information of each product line of the order.
- *Payments*: Stores payments made by customers based on their accounts.
- *Employees*: Store information about the employees of the organization.
- *Offices*: Stores sales office data.

The figure below illustrates the relationship between tables in the database.

productlines
- 🔑 productLine VARCHAR(50)
- ◇ textDescription VARCHAR(4000)
- ◇ htmlDescription MEDIUMTEXT
- ◇ image MEDIUMBLOB

Indexes

products
- 🔑 productCode VARCHAR(15)
- ◇ productName VARCHAR(70)
- 🔴 productLine VARCHAR(50)
- ◇ productScale VARCHAR(10)
- ◇ productVendor VARCHAR(50)
- ◇ productDescription TEXT
- ◇ quantityInStock SMALLINT(6)
- ◇ buyPrice DOUBLE
- ◇ MSRP DOUBLE

Indexes

orderdetails
- 🔑 orderNumber INT(11)
- 🔑 productCode VARCHAR(15)
- ◇ quantityOrdered INT(11)
- ◇ priceEach DOUBLE
- ◇ orderLineNumber SMALLINT(6)

Indexes

employees
- 🔑 employeeNumber INT(11)
- ◇ lastName VARCHAR(50)
- ◇ firstName VARCHAR(50)
- ◇ extension VARCHAR(10)
- ◇ email VARCHAR(100)
- 🔴 officeCode VARCHAR(10)
- ◇ reportsTo INT(11)
- ◇ jobTitle VARCHAR(50)

Indexes

customers
- 🔑 customerNumber INT(11)
- ◇ customerName VARCHAR(50)
- ◇ contactLastName VARCHAR(50)
- ◇ contactFirstName VARCHAR(50)
- ◇ phone VARCHAR(50)
- ◇ addressLine1 VARCHAR(50)
- ◇ addressLine2 VARCHAR(50)
- ◇ city VARCHAR(50)
- ◇ state VARCHAR(50)
- ◇ postalCode VARCHAR(15)
- ◇ country VARCHAR(50)
- 🔴 salesRepEmployeeNumber INT(11)
- ◇ creditLimit DOUBLE

Indexes

orders
- 🔑 orderNumber INT(11)
- ◇ orderDate DATETIME
- ◇ requiredDate DATETIME
- ◇ shippedDate DATETIME
- ◇ status VARCHAR(15)
- ◇ comments TEXT
- 🔴 customerNumber INT(11)

Indexes

offices
- 🔑 officeCode VARCHAR(10)
- ◇ city VARCHAR(50)
- ◇ phone VARCHAR(50)
- ◇ addressLine1 VARCHAR(50)
- ◇ addressLine2 VARCHAR(50)
- ◇ state VARCHAR(50)
- ◇ country VARCHAR(50)
- ◇ postalCode VARCHAR(15)
- ◇ territory VARCHAR(10)

Indexes

payments
- 🔑 customerNumber INT(11)
- 🔑 checkNumber VARCHAR(50)
- ◇ paymentDate DATETIME
- ◇ amount DOUBLE

Indexes

Download *mysqlsampledatabase.sql* file script to create the database from the address:

Assuming the *mysqlsampledatabase.sql* file is located in the root of Drive C:

Log in to the MySQL server from the *mysql.exe* client using the root account

From the mysql> prompt, execute the following statement:

```
source C:\mysqlsampledatabase.sql
```

Database is created named ***classicmodels***

2. Execute the query with the SELECT statement

SELECT syntax

```
SELECT column 1, column 2, ...
FROM tables
[WHERE selected conditions]
[GROUP BY group]
[HAVING group selection conditions]
[ORDER BY sorted columns]
[LIMIT quantity limit];
```

- In a `SELECT` query, there are many optional elements that can be used. The options are enclosed in square brackets [].
- The order of the keywords `WHERE`, `GROUP BY`, `HAVING`, `ORDER BY` and `LIMIT` must be in the above order.

To select all columns in a table, use the asterisk (*) symbol instead of listing all column names after the `SELECT` keyword.

Example: If you need to query all employee information, you can use the following query:

```
SELECT * FROM employees;
```

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1	NULL	President
1056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1	1002	VP Sales
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
1143	Bow	Anthony	x5428	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	Firrelli	Julie	x2173	jfirrelli@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
1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep

It is also possible to view partial data of a table by listing the names of the columns after the `SELECT` keyword. This is called *projection*.

Example: If you need to get the *employee's first name, last name, job title* you can use the following query:

```
SELECT lastname, firstname, jobtitle
FROM employees;
```

lastname	firstname	jobtitle
Murphy	Diane	President
Patterson	Mary	VP Sales
Firrelli	Jeff	VP Marketing
Patterson	William	Sales Manager (APAC)
Bondur	Gerard	Sale Manager (EMEA)
Bow	Anthony	Sales Manager (NA)
Jennings	Leslie	Sales Rep
Thompson	Leslie	Sales Rep
Firrelli	Julie	Sales Rep
Patterson	Steve	Sales Rep
Tseng	Foon Yue	Sales Rep
Vanauf	George	Sales Rep
Bondur	Loui	Sales Rep
Hernandez	Gerard	Sales Rep
Castillo	Pamela	Sales Rep

Example: To retrieve information about *Product Codes* and *Product Names*, execute the following query:

```
SELECT ProductCode, ProductName
FROM Products;
```

ProductCode	ProductName
S10_1678	1969 Harley Davidson Ultimate Chopper
S10_1949	1952 Alpine Renault 1300
S10_2016	1996 Moto Guzzi 1100i
S10_4698	2003 Harley-Davidson Eagle Drag Bike
S10_4757	1972 Alfa Romeo GTA
S10_4962	1962 LanciaA Delta 16V
S12_1099	1968 Ford Mustang
S12_1108	2001 Ferrari Enzo
S12_1666	1958 Setra Bus
S12_2823	2002 Suzuki XREO
S12_3148	1969 Corvair Monza
S12_3380	1968 Dodge Charger
S12_3891	1969 Ford Falcon
S12_3990	1970 Plymouth Hemi Cuda
S12_4473	1957 Chevy Pickup

3. WHERE clause

The WHERE clause of the SELECT statement allows you to select specific rows that match the search condition or criteria. Use the WHERE clause to filter records based on a certain condition.

Example: Company president's information can be found by the following query:

```
SELECT firstName, lastName, email
FROM employees
WHERE jobtitle = "President";
```

firstname	lastname	email
Diane	Murphy	dmurphy@classicmodelcars.com

Or you can find out the information about the customer number *112* by the following query:

```
SELECT *
FROM customers
WHERE customerNumber=112;
```

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.

Example: Get information about the orders of the customer number 181:

```
SELECT *
FROM orders
WHERE customerNumber = 181;
```

orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber
10102	2003-01-10 00:00:00	2003-01-18 00:00:00	2003-01-14 00:00:00	Shipped	NULL	181
10237	2004-04-05 00:00:00	2004-04-12 00:00:00	2004-04-10 00:00:00	Shipped	NULL	181
10324	2004-11-05 00:00:00	2004-11-11 00:00:00	2004-11-08 00:00:00	Shipped	NULL	181
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Combine conditions with AND and OR operators

We can combine two or more different conditions in the WHERE clause, using the AND and OR operators. For two conditions joined by AND, both need to be true for a join condition to be true. For two conditions joined by OR, the combination condition is true if one or both conditions are true.

Example: Give the information about the customer in the USA of the Sales Representative Number 1165.

```
SELECT *
FROM customers
WHERE country ='USA' AND salesRepEmployeeNumber = 1165;
```

contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber
Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA	1165
Julie	6505555787	5557 North Pendale Street	NULL	San Francisco	CA	94217	USA	1165
Juri	6505556809	9408 Furth Circle	NULL	Burlingame	CA	94217	USA	1165
Julie	6505551386	7734 Strong St.	NULL	San Francisco	CA	94217	USA	1165
Sue	4085553659	3086 Ingle Ln.	NULL	San Jose	CA	94217	USA	1165
Sue	4155554312	2793 Furth Circle	NULL	Brisbane	CA	94217	USA	1165
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Example: Give the information about the orders with a status 'On Hold' or 'In Process'

```
SELECT *
```

```
FROM orders
WHERE status = 'On Hold' or status ='In Process';
```

orderNumber	orderDate	requiredDate	shippedDate	status	comments
10334	2004-11-19 00:00:00	2004-11-28 00:00:00	NULL	On Hold	The outstaniding balance for this customer exceeds their credit limit. Order will
10401	2005-04-03 00:00:00	2005-04-14 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
10407	2005-04-22 00:00:00	2005-05-04 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
10414	2005-05-06 00:00:00	2005-05-13 00:00:00	NULL	On Hold	Customer credit limit exceeded. Will ship when a payment is received.
10420	2005-05-29 00:00:00	2005-06-07 00:00:00	NULL	In Process	NULL
10421	2005-05-29 00:00:00	2005-06-06 00:00:00	NULL	In Process	Custom shipping instructions were sent to warehouse
10422	2005-05-30 00:00:00	2005-06-11 00:00:00	NULL	In Process	NULL
10423	2005-05-30 00:00:00	2005-06-05 00:00:00	NULL	In Process	NULL
10424	2005-05-31 00:00:00	2005-06-08 00:00:00	NULL	In Process	NULL
10425	2005-05-31 00:00:00	2005-06-07 00:00:00	NULL	In Process	NULL

5. IS NULL: find NULL values

For fields that have undefined value yet, SQL consider that value as NULL. To check if a field has a NULL value or not, instead of using the = comparison, SQL use the operation *IS NULL*.

Example: Given customers who have NOT been assigned by a Sales Representative.

```
SELECT customerName, salesRepEmployeeNumber

FROM customers

WHERE salesRepEmployeeNumber = NULL;
```

If using the = comparison as above, no row will be returned. If instead of using = comparison, we use IS NULL:

```
SELECT customerName, salesRepEmployeeNumber

FROM customers

WHERE salesRepEmployeeNumber IS NULL;
```

customerName	salesRepEmployeeNumber
Havel & Zbyszek Co	NULL
Porto Imports Co.	NULL
Asian Shopping Network, Co	NULL
Natürlich Autos	NULL
ANG Resellers	NULL
Messner Shopping Network	NULL
Franken Gifts, Co	NULL
BG&E Collectables	NULL
Schuyler Imports	NULL
Der Hund Imports	NULL
Cramer Spezialitäten, Ltd	NULL
Asian Treasures, Inc.	NULL
SAR Distributors, Co	NULL
Kommission Auto	NULL
Lisboa Souveniers, Inc	NULL
Stuttgart Collectable Exch...	NULL

6. DISTINCT keyword

With the `DISTINCT` keyword, it is possible to remove duplicate rows from the `SELECT` statement.

Example: To find how many *Job Title* of all employees, use the `DISTINCT` keyword in the `SELECT` statement as follows:

```
SELECT DISTINCT jobTitle FROM employees;
```

	jobTitle
▶	President
	VP Sales
	VP Marketing
	Sales Manager (APAC)
	Sale Manager (EMEA)
	Sales Manager (NA)
	Sales Rep

Or you can find out the *Product Codes* purchased by the query as follows:

```
SELECT DISTINCT productCode FROM orderDetails;
```

	productCode
▶	S18_1749
	S18_2248
	S18_4409
	S24_3969
	S18_2325
	S18_2795
	S24_1937
	S24_2022
	S18_1342
	S18_1367
	S10_1949
	S10_4962
	S12_1666
	S18_1097
	S18 2432

7. Limiting query results by using LIMIT statement

In most queries, when working with data tables that have thousands to millions of records and we do not want to write a query to get all that data to ensure performance and traffic between

database servers and application servers. MySQL supports a feature of `LIMIT` which allows limiting the number of rows returned with the `SELECT` statement.

Assuming we have a database table with 10,000 records and want to get the first N records, we can use the following query:

```
SELECT * FROM table_name
LIMIT N;
```

We can also use LIMIT to retrieve a specified number of records from a specified position:

```
LIMIT S, N;
```

In the above query, S is the starting point for index. MySQL determines that the first position to be recorded begins with 0; N is the number of records you want to select.

Example: Given the information about the names of the first five products in the Product table by using the following query:

```
SELECT productName FROM products
LIMIT 5;
```

productName
1969 Harley Davidson Ultimate Chopper
1952 Alpine Renault 1300
1996 Moto Guzzi 1100i
2003 Harley-Davidson Eagle Drag Bike
1972 Alfa Romeo GTA

Or we can retrieve information about the first ten customers currently in Framce by using the following query:

```
SELECT * FROM customers
WHERE country='France'
LIMIT 10;
```

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	address
103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	NULL
119	La Rochelle Gifts	Labrune	Janine	40.67.8555	67, rue des Cinquante Otages	NULL
146	Saveley & Henriot, Co.	Saveley	Mary	78.32.5555	2, rue du Commerce	NULL
171	Daedalus Designs Imports	Rancé	Martine	20.16.1555	184, chaussée de Tournai	NULL
172	La Corne D'abondance, Co.	Bertrand	Marie	(1) 42.34.2555	265, boulevard Charonne	NULL
209	Mini Caravy	Citeaux	Frédérique	88.60.1555	24, place Kléber	NULL
242	Alpha Cognac	Roulet	Annette	61.77.6555	1 rue Alsace-Lorraine	NULL
250	Lyon Souveniers	Da Silva	Daniel	+33 1 46 62 7555	27 rue du Colonel Pierre Avia	NULL
256	Auto Associés & Cie.	Tonini	Daniel	30.59.8555	67, avenue de l'Europe	NULL
350	Marseille Mini Autos	Lebihan	Laurence	91.24.4555	12, rue des Bouchers	NULL

❖ Practical exercises:

1. Get a list of employees with unspecified reportsTo field.
2. Get a list of CustomerNumber that has made transactions.
3. Get a list of orders that have shipped on 18 Jan, 2003. *Note:* MySQL stores datetime data in the format of year/month/day (YYYY/MM/DD).
4. Get a list of orders with the date placed in April 2005 and have status of 'Shipped'.
5. Get a list of products that are in the 'Classic Cars' category.