

SECTION1: Overview

<https://www.udemy.com/sql-mysql-for-data-analytics-and-business-intelligence/learn/lecture/8337984#overview>

SQL = Structured Query Language

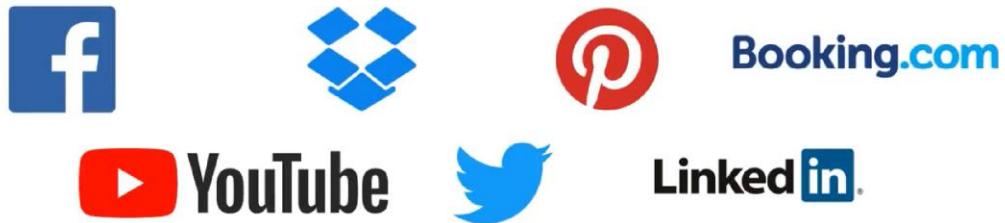
- 1. a programming language specifically designed for working with databases
- create
- manipulate **DATA**
- share from Relational Database Management Systems

- 1. a programming language specifically designed for working with databases
- 2. allows you to write queries that the computer can execute and then provide database insights in return

选择 Database management system! 不同 system 之间区别不大。

市面上很多使用 SQL 语言的平台，但是 MySQL 比较好：

1. Open-source/免费
2. 使用排名靠前
3. Reliable, mature



DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



334 systems in ranking, September 2017

Rank	Sep 2017	Aug 2017	Sep 2016	DBMS	Database Model	Score		
						Sep 2017	Aug 2017	Sep 2016
1.	1.	1.	1.	Oracle	Relational DBMS	1359.09	-8.78	-66.47
2.	2.	2.	2.	MySQL	Relational DBMS	1312.61	-27.69	-41.41
3.	3.	3.	3.	Microsoft SQL Server	Relational DBMS	1212.54	-12.93	+0.99
4.	4.	4.	4.	PostgreSQL	Relational DBMS	372.36	+2.60	+56.01
5.	5.	5.	5.	MongoDB	Document store	332.73	+2.24	+16.74
6.	6.	6.	6.	DB2	Relational DBMS	198.34	+0.87	+17.15
7.	7.	↑ 8.	8.	Microsoft Access	Relational DBMS	128.81	+1.78	+5.50
8.	8.	↓ 7.	7.	Cassandra	Wide column store	126.20	-0.52	-4.29
9.	9.	↑ 10.	10.	Redis	Key-value store	120.41	-1.49	+12.61
10.	10.	↑ 11.	11.	Elasticsearch	Search engine	120.00	+2.35	+23.52

SQL 基于 relational database

Relational database: record*field, a few related tables form relational database

如果把所有数据放在一个 table 里, 有太多列, 人们不能很好的了解信息, 所以按照主题放在不同 table 中

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2016	1	A_1
2	02/12/2016	2	C_1
3	15/04/2017	3	D_1
4	24/05/2017	1	B_2
5	25/05/2017	4	B_2
6	06/06/2017	2	B_1
7	10/06/2017	4	A_2
8	13/06/2017	3	C_1
9	20/07/2017	1	A_1
10	11/08/2017	2	B_1

Items					
item_code	item	unit_price_usd	company_id	company	headquarters_phone_number
A_1	Lamp	20	1	Company A	+1 (202) 555-0196
A_2	Desk	250	1	Company A	+1 (202) 555-0196
B_1	Lamp	30	2	Company B	+1 (202) 555-0152
B_2	Desk	350	2	Company B	+1 (202) 555-0152
C_1	Chair	150	3	Company C	+1 (229) 853-9913
D_1	Loudspeakers	400	4	Company D	+1 (618) 369-7392

relational algebra allows us to retrieve data efficiently

一个最小数据单位/表: entity

entity = the smallest unit that can contain a meaningful set of data

horizontal entity

SALES			
purchase_number	date_of_purchase	customer_id	item_code
1	03/09/2016	1	A_1
2	02/12/2016	2	C_1
3	15/04/2017	3	D_1
4	24/05/2017	1	B_2
5	25/05/2017	4	B_2
6	06/06/2017	2	B_1
7	10/06/2017	4	A_2
8	13/06/2017	3	C_1
9	20/07/2017	1	A_1
10	11/08/2017	2	B_1

vertical entity

Section 2:

SQL: create & manipulate relational databases

types of programming:

procedural (imperative)

object-oriented

SQL

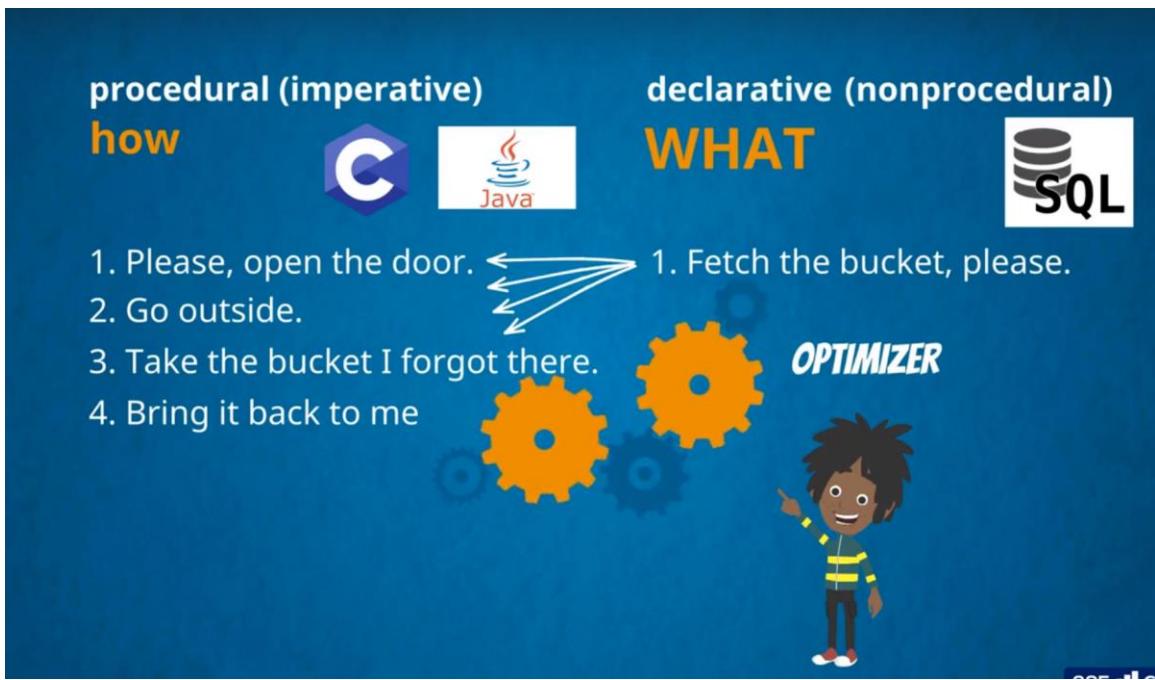
declarative (nonprocedural)

functional



Procedural 就是要好几步,

Declarative 一般一步到位, 直接呈现结果 (其实有 built in optimizer 已经帮我们分好步自行运行了) 所以关注结果/想要什么就行



main components of SQL's syntax:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)

创建对象：DDL,一般是 create

Data Definition Language

- Data Definition Language (DDL)

- CREATE
- ALTER
- DROP
- RENAME
- TRUNCATE

Data Definition Language

- SQL's syntax

comprises several types of statements that allow you to perform various commands and operations

- Data Definition Language (DDL)

- a syntax
- a set of statements that allow the user to define or modify data structures and objects, such as tables

- the CREATE statement

used for creating entire databases and database objects as tables

Data Definition Language

- the CREATE statement

used for creating entire databases and database objects as tables



CREATE [object_type] object_name;

SQL

CREATE TABLE object_name (column_name data_type);

Data Definition Language

- the ALTER statement

used when altering existing objects

- ADD
- REMOVE
- RENAME

Data Definition Language



```
ALTER TABLE sales  
ADD COLUMN date_of_purchase DATE;
```

sales	
purchase_number	date_of_purchase

Data Definition Language



```
DROP object_type object_name;
```

```
DROP TABLE customers;
```

customers

customer_id	first_name

Data Definition Language



```
RENAME object_type object_name TO new_object_name;  
RENAME TABLE customers TO customer_data;
```

customer_id	first_name

Data Definition Language



```
TRUNCATE object_type object_name;  
TRUNCATE TABLE customers;
```

customers	
customer_id	first_name

命名 var 时不能用函数

Data Manipulation Language

- Data Manipulation Language (DML)

its statements allow us to manipulate the data in the tables of a database

- the SELECT statement

used to retrieve data from database objects, like tables

Data Manipulation Language

`SELECT... FROM sales...;`



SQL

sales	
	purchase_number



Data Manipulation Language

`INSERT INTO sales (purchase_number, date_of_purchase) VALUES
(1, '2017-10-11');`



SQL

sales	
purchase_number	date_of_purchase

Data Manipulation Language



```
SQL  
UPDATE sales  
SET date_of_purchase = '2017-12-12'  
WHERE purchase_number = 1;
```

sales	
purchase_number	date_of_purchase
1	2017-10-11
2	2017-10-27

Truncate 删所有行

Delete 可以 customize

Data Manipulation Language

- the DELETE statement

- functions similarly to the TRUNCATE statement

- TRUNCATE vs. DELETE

- TRUNCATE allows us to remove all the records contained in a table

- vs.

- with DELETE, you can specify precisely what you would like to be removed

Data Manipulation Language



```
DELETE FROM sales  
WHERE  
    purchase_number = 1;
```

SQL

sales	
purchase_number	date_of_purchase
2	2017-10-27

Data Manipulation Language

- Data Manipulation Language (DML)

- SELECT... FROM...
- INSERT INTO... VALUES
- UPDATE... SET... WHERE
- DELETE... FROM... WHERE

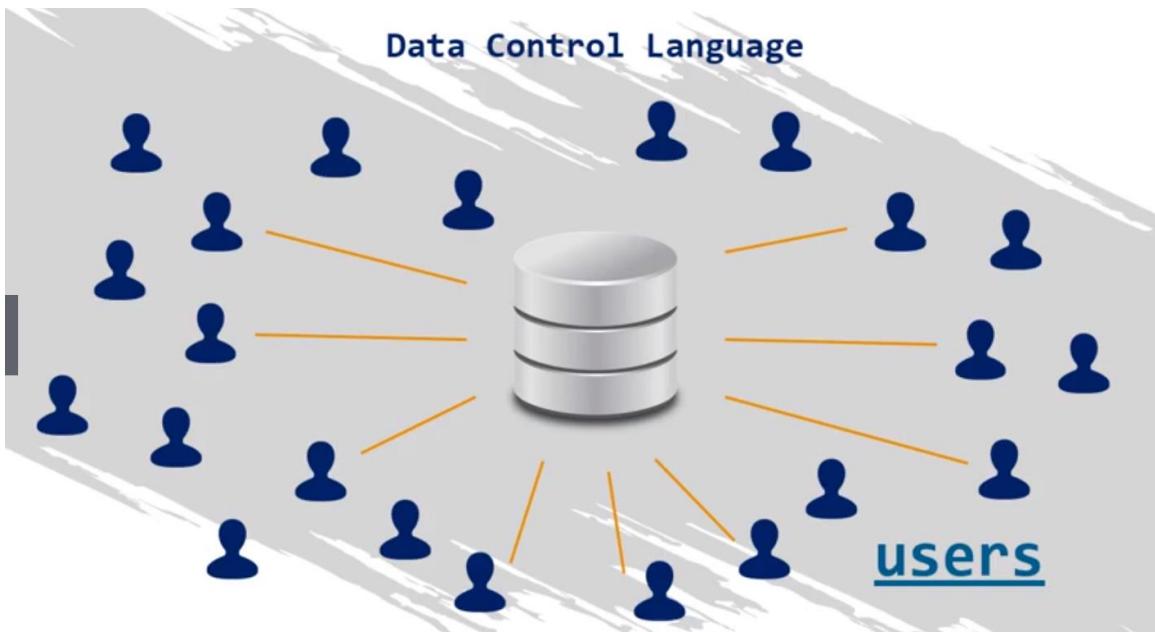
Data Control Language

- Data Control Language (DCL)

- the GRANT and REVOKE statements

allow us to manage the rights users have in a database

User may have dif right to access



Grant 给某人 弄某表的某级别权限

Data Control Language

- The GRANT statement
 - gives (or grants) certain permissions to users
 - one can grant a *specific type* of permission, like *complete or partial access*

SQL

```
GRANT type_of_permission ON database_name.table_name TO
  'username'@'localhost'
```

The server is based on our machine and it's commonly known as "localhost", which is the domain name for the local IP address of our machine 127.0.0.1.

Naturally, big companies and corporations don't use this type of server and their databases lay on external, much more powerful servers specifically designed to store big amounts of data.

Those servers are not our "localhost" and are accessible on different IP addresses.

Data Control Language

- these rights will be assigned to a person who has a *username* registered at the *Local server* ('*localhost*': IP 127.0.0.1)
- big companies and corporations don't use this type of server, and their databases lay on external, more powerful servers



```
GRANT type_of_permission ON database_name.table_name TO  
'username'@'localhost'
```

SQL

Create user, 密码为 pass

```
CREATE USER 'frank'@'localhost' IDENTIFIED BY 'pass';
```

给这个人的权限是 select, 及不能用右侧的, 只能看到 sales 中的 customer

```
GRANT SELECT ON sales.customers TO 'frank'@'localhost';
```

DROP

```
GRANT ALL ON Sales.* to 'frank'@'localhost';
```

TRUNCATE

ALTER

INSERT

DELETE

Administrator 有权授权 grant/撤权 revoke

Data Control Language

- Database administrators

people who have *complete* rights to a database

- they can grant access to users and can revoke it

- the REVOKE clause

used to revoke permissions and privileges of database users

- the exact opposite of GRANT

```
REVOKE SELECT ON sales.customers FROM 'frank'@'localhost';
```

Transaction Control Language

- Transaction Control Language (DCL)

- not every change you make to a database is saved automatically

- the COMMIT statement

- related to INSERT, DELETE, UPDATE

- will save the changes you've made

- will let other users have access to the modified version of the database

若某 user 更新数据库，如何全面更新，让其他 user 看到更新后的

Transaction Control Language

DB administrator

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

Problem:

users



142 people bookmarked this moment.

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

Transaction Control Language

DB administrator



```
UPDATE customers  
SET last_name = 'Johnson'  
WHERE customer_id = 4  
COMMIT;
```

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

DB administrator

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

users

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Johnson	c.winnfield@365careers.com	0	

每次都点 commit 太麻烦

Transaction Control Language

- the COMMIT statement
committed states can accrue
- the ROLLBACK clause
the clause that will let you make a step back
 - allows you to undo any changes you have made but don't want to be saved permanently

Roll back : revert to last non-commit state (不太清楚区别, 可以看看 Database 书)

DB administrator

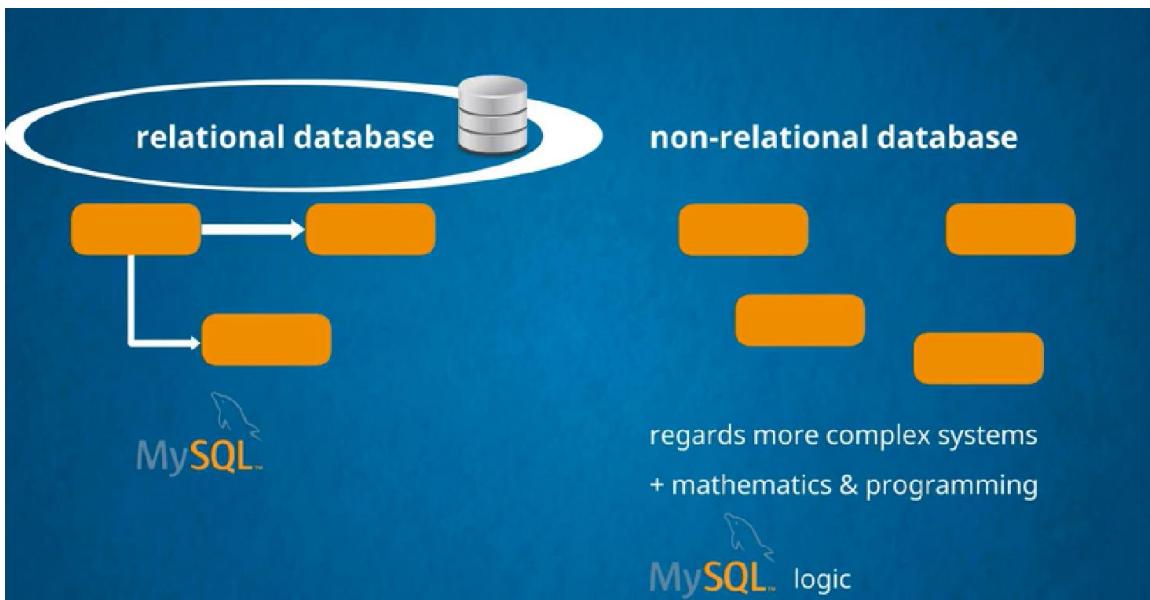
```
UPDATE customers
SET last_name = 'Johnson'
WHERE customer_id = 4
COMMIT;

ROLLBACK;
```

 SQL

Customers					
customer_id	first_name	last_name	email_address	number_of_complaints	
1	John	McKinley	john.mackinley@365careers.com	0	
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2	
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1	
4	Catherine	Winnfield	c.winnfield@365careers.com	0	

SECTION3: Relational Database



relational databases



Spreadsheets

provide a stable structure,
controlling access permissions
and user restrictions



google docs



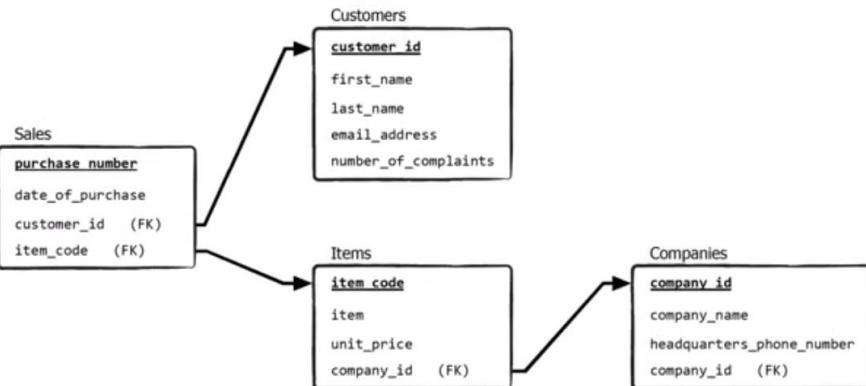
Problem:

finding out who changed or
deleted information incorrectly

Entity-Relationship (ER) diagram



Database Schema



152 people bookmarked this moment.

database design + creation + manipulation



database design + creation + manipulation

= database management

database administration

daily maintenance for database – administration

Primary key: 此表的 unique 区分每行的东西，可以是两列的组合 unique

Relational Schemas: Primary Key

Sales

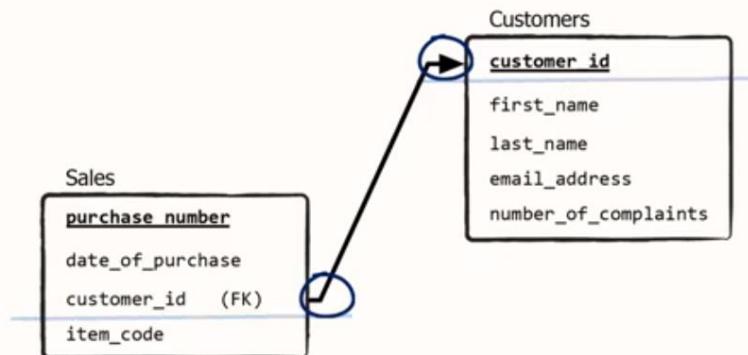
<u>purchase_number</u>
date_of_purchase
customer_id (FK)
item_code (FK)

Table name: Sales
Primary key: purchase_number
Other fields: date_of_purchase, customer_id, item_code

Foreign key 此表连到其他表的（箭头指向其他表）自己表内的 FK 可以重复（多对一）

Foreign key shows the relationships of databases

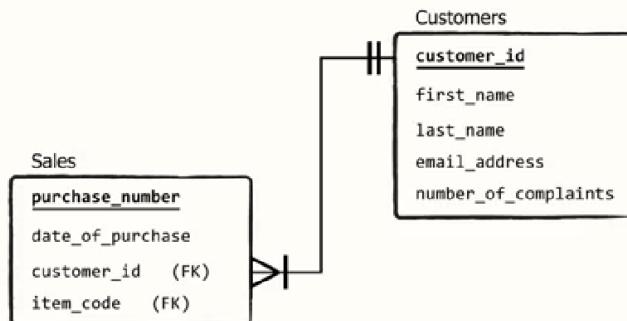
Relational Schemas: Foreign Key

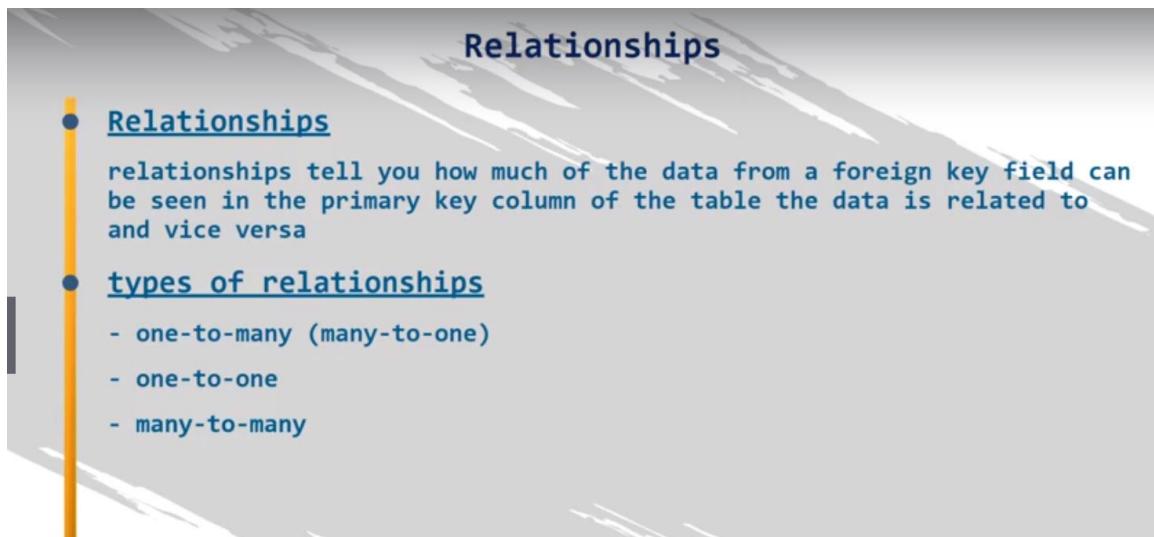


Relationships

Customers to Sales: one-to-many

Sales to Customers: many-to-one





SECTION4: Install MySQL and get started

Mysql connection – 加号—新加 connection

可以 load multiple connections when using workbench

The screenshot shows a Command Prompt window titled "Command Prompt - mysql -u root -p". The window contains the following text:

```
C:\Users\365>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'nativeuser'@'localhost'
      -> IDENTIFIED WITH mysql_native_password BY '365Pass';
Query OK, 0 rows affected (0.19 sec)

mysql>
```



```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'nativeuser'@'localhost';
Query OK, 0 rows affected (0.14 sec)

mysql> -
```

我的电脑里：“C:\Program Files\MySQL\MySQL Server 8.0\bin”

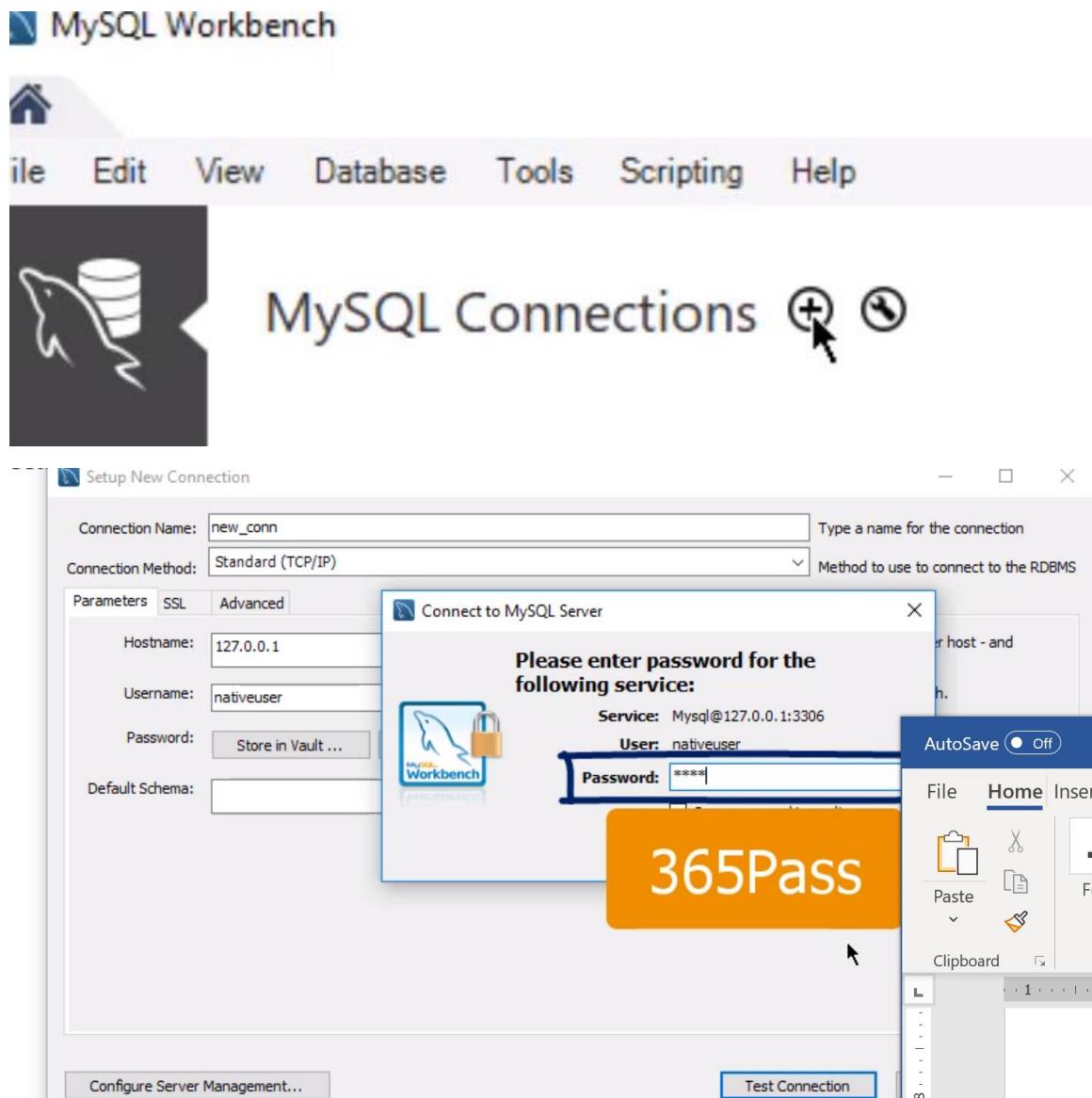
在 prompt 中输入

```
cd C:\Program Files\MySQL\MySQL Server 8.0\bin
```

```
CREATE USER 'user2'@'localhost'  
IDENTIFIED WITH mysql_native_password BY 'user2';
```

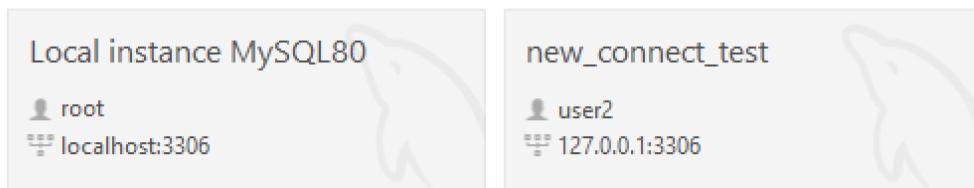
```
GRANT ALL PRIVILEGES ON *.* TO 'user2'@'localhost';
```

然后回到 sql 界面点这里建新的连接



第一行随便填，后面都是按照设置的

MySQL Connections



熟悉操作界面

The screenshot shows the MySQL Workbench interface with the following components visible:

- Navigator:** On the left, it includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore) and INSTANCE (Startup / Shutdown, Server Logs, Options File).
- Query Editor:** On the right, it has tabs for "Query 1" and "Query 2". The "Query 1" tab is active, showing the SQL query: `select * from employees;`. A red circle highlights the folder icon in the toolbar above the editor.

这个是直接打开一个新的 sql 文件

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL57' is selected. The 'Query 1' tab contains the SQL command: 'select * from employees;'. Below the command is a result grid showing data from the 'employees' table. The columns are: emp_no, birth_date, first_name, last_name, gender, and hire_date. The data includes rows from 10001 to 10008. The 'Result Grid' tab is active. On the right side, there's a 'SQLAdditions' panel with a note about automatic context help. The bottom status bar shows '1000 row(s) returned'.

关掉一个 output

Section 5

String Data Types

<u>string data type</u>	<u>Storage</u>	<u>Example</u>	<u>Length</u> (symbols)	<u>size</u> (bytes)
character	CHAR	fixed	CHAR(5)	
			'James'	5
			'Bob'	5
variable character	VARCHAR	variable	VARCHAR(5)	
			'James'	5
			'Bob'	3

String Data Types

<u>string data type</u>		<u>Maximum size (bytes)</u>	
<i>character</i>	CHAR	255	50% faster
<i>variable character</i>	VARCHAR	65,535	a lot more responsive to the data value inserted

固定的长度: char(3)是刚刚好的三个 digit

String Data Types

Companies		
company_id	headquarters_phone_number	company
1	+1 (202) 555-0196	COA
2	+1 (202) 555-0152	COB
3	+1 (229) 853-9913	COC
4	+1 (618) 369-7392	COD

company **CHAR(3)**

Password:

the symbols cannot be more than
10 characters

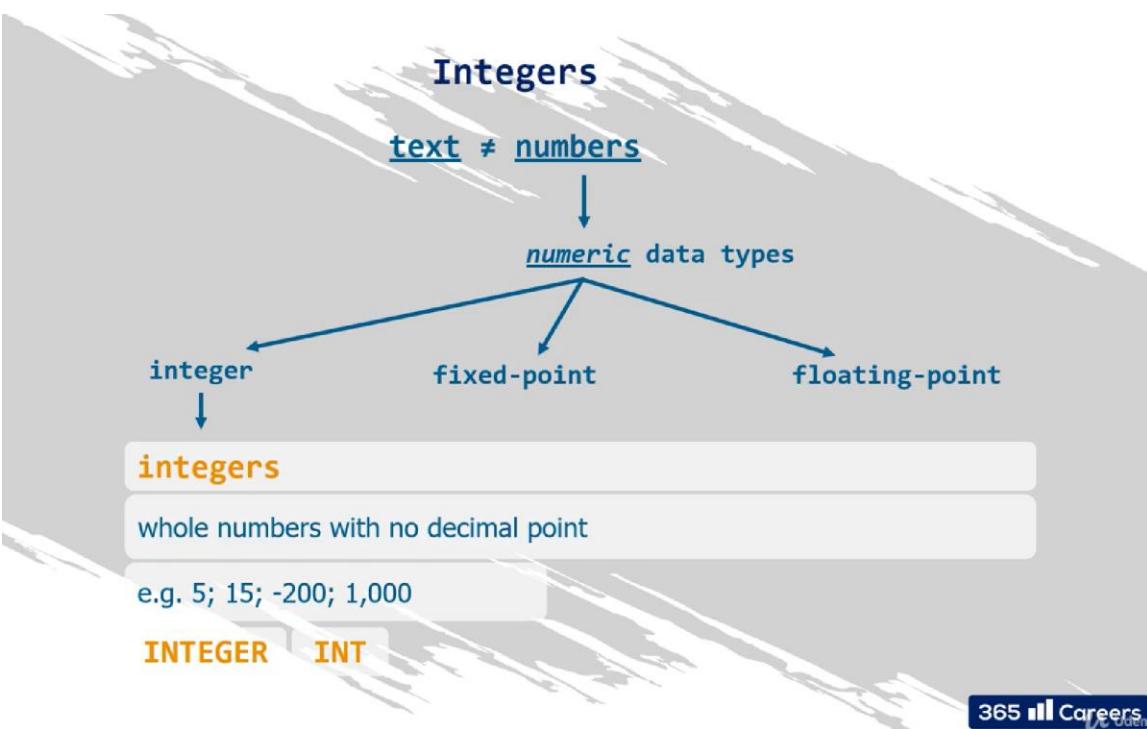
password **VARCHAR(10)**

一共有三种 string data type, 最后 enum 是枚举, 列出能够在这里接受的所有 string

String Data Types

<u>string data type</u>		<u>Example</u>
character	CHAR	CHAR(5)
variable character	VARCHAR	VARCHAR(5)
ENUM ("enumerate")	ENUM	ENUM('M', 'F') ERROR MySQL will show an error if you attempt to insert any value different from "M" or "F".

Integers



Integers

<u>numeric data type</u>	<u>size (bytes)</u>	<u>minimum value</u> (signed/unsigned)	<u>maximum value</u> (signed/unsigned)
TINYINT	1	-128 0	127 255
SMALLINT	2	-32,768 0	32,767 65,535
MEDIUMINT	3	-8,388,608 0	8,388,607 16,777,215
INT	4	-2,147,483,648 0	2,147,483,647 4,294,967,295
BIGINT	8	-9,223,372,036,854,775,808 0	9,223,372,036,854,775,807 18,446,744,073,709,551,615

SIGNED: 正负值都可以

Unsigned: 只能是非负数

一般是 signed, 若想 unsigned 要定义

选择刚好好的 string 定义以降低 store space

Decimal(总 digit 数, 小数后 digit 数)

Decimal(7) 与 decimal(7,0)等价

Fixed- and Floating-Point Data Types

- fixed-point data represent exact values

DECIMAL (5 , 3)

10.523

10.5

10.500

10.5236789

10.524



Fixed- and Floating-Point Data Types

• DECIMAL = NUMERIC

e.g. salaries

NUMERIC (p , s)

precision: p = 7

scale: s = 2

e.g. NUMERIC (7,2) \$ 75,000.50

Float 可以入四舍五入近似, fixed 是确切值

Fixed- and Floating-Point Data Types

the main difference between the fixed- and the floating-point type is in the way the value is represented in the memory of the computer

DECIMAL (5 , 3)

10.5236789

10.524



fixed

FLOAT (5 , 3)

10.5236789

10.524



floating

Fixed- and Floating-Point Data Types

<u>Floating-point data type</u>	<u>size (bytes)</u>	<u>precision</u>	<u>maximum number of digits</u>
FLOAT	4	<i>single</i>	23
DOUBLE	8	<i>double</i>	53

Other Useful Data Types

DATE



used to represent a date in the format YYYY-MM-DD

1st of January 1000 - 31st of December 9999

e.g. 25th of July 2018: '2018-07-25'

Other Useful Data Types

DATE



= DATETIME

next to the date, we could save the time:

YYYY-MM-DD HH:MM:SS [.fraction]

0 - 23:59:59.999999

e.g. 25th of July 2018 9:30 a.m.: '2018-07-25 9:30:00'

Other Useful Data Types

DATETIME

represents the date shown on the calendar and the time shown on the clock

vs.

TIMESTAMP

used for a *well-defined, exact point in time*

Other Useful Data Types

TIMESTAMP

used for a *well-defined, exact point in time*

1st of January 1970 UTC – 19th of January 2038, 03:14:07 UTC

- records the moment in time as the number of seconds passed after the 1st of January 1970 00:00:00 UTC

e.g. 25th of July 2018:

1,535,155,200

more than 48 years!

Other Useful Data Types

TIMESTAMP

- representing a moment in time as a number allows you to easily obtain the difference between two TIMESTAMP values

e.g. end time:

'2018-07-25 10:30:00' UTC

TIMESTAMP

start time:

'2018-07-25 09:00:00' UTC

TIMESTAMP

5,400

TIMESTAMP

Other Useful Data Types	
string, date, and time data types	numeric data types
CHAR	INTEGER
VARCHAR	DECIMAL
DATE	NUMERIC
DATETIME	FLOAT
TIMESTAMP	DOUBLE
data must be written within quotes	
only numeric values are written without quotes	

存文件/大文件

Other Useful Data Types				
BLOB	Binary Large OBject			
	<ul style="list-style-type: none"> - refers to a file of binary data - data with 1s and 0s - involves saving files in a record 			
 *.doc	 *.xlsx	 *.xml	 *.jpg	 *.wav

Other Useful Data Types

Customers						
customer_id	first_name	last_name	email_address	number_of_complaints	photo	
1	John	McKinley	john.mackinley@365careers.com	0		
2	Elizabeth	McFarlane	e.mcfarlane@365careers.com	2		
3	Kevin	Lawrence	kevin.lawrence@365careers.com	1		
4	Catherine	Winnfield	c.winnfield@365careers.com	0	*.jpg	

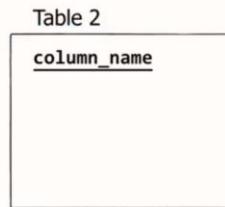
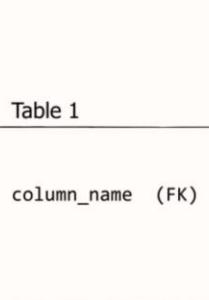
Section 6

Can see ddl of the table

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the connection name 'new_connect_test' is selected. Below the menu bar, the 'DDL' tab is active. On the left, the 'Navigator' pane shows the database structure with 'SCHEMAS' expanded to include 'optprojed', 'project', 'sakila', and 'sales'. Under 'sales', 'Tables' is expanded to show 'sales'. The main pane displays the DDL code for the 'sales' table:

```
1 CREATE TABLE `sales` (
2     `purchase_number` int(11) NOT NULL AUTO_INCREMENT,
3     `date_of_purchase` date NOT NULL,
4     `customer_id` int(11) DEFAULT NULL,
5     `item_code` varchar(10) NOT NULL,
6     PRIMARY KEY (`purchase_number`)
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

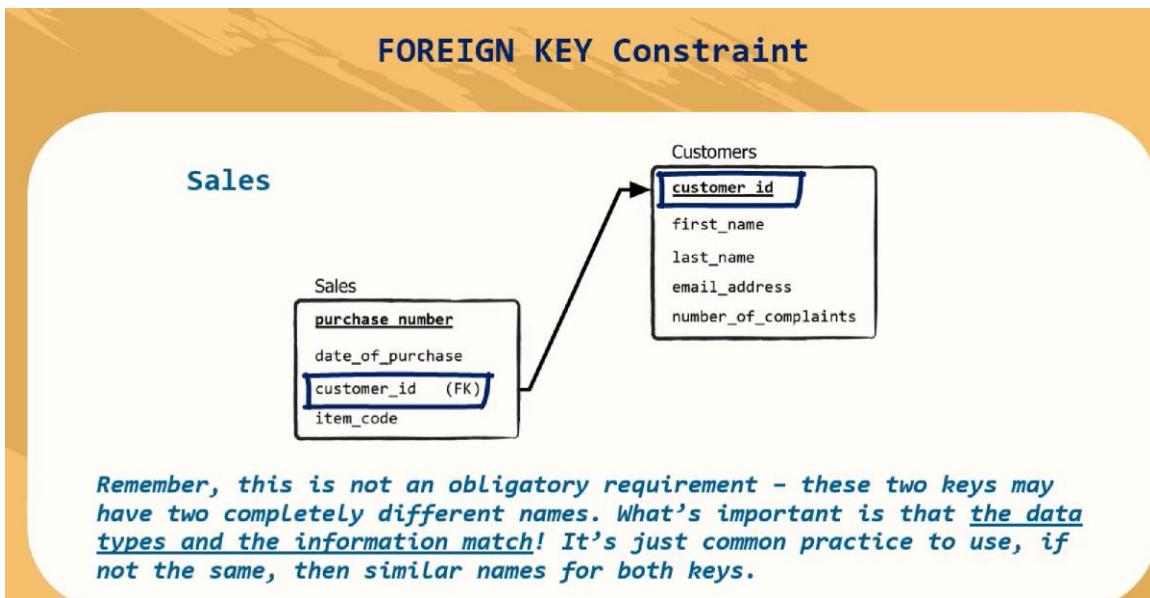
FOREIGN KEY Constraint



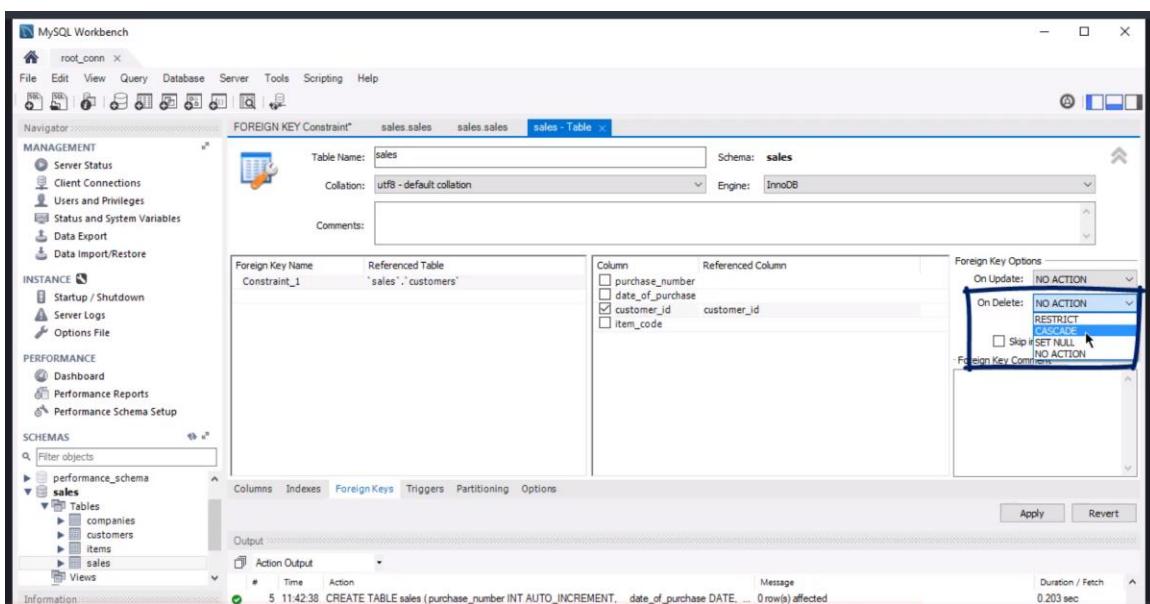
parent table = referenced table

child table = referencing table

FOREIGN KEY Constraint



不想写代码可以之间右键 table-alter table 等



NOT NULL Constraint

- Don't confuse a NULL value with the value of 0 or with a "NONE" response!

Think of a null value as a missing value.

0	NONE	NULL
assigned by the user		assigned by the computer

Section 7

Coding Techniques and Best Practices

clean code

code that is *focused* and *understandable*, which means it must be readable, logical, and changeable

Coding Techniques and Best Practices

- when assigning names to variables or SQL objects,
always chose shorter, meaningful names, conveying specific information

↓
pronounceable, where one word per concept has been picked



Sales				
purchase_number	date_of_purchase	customer_id	item_code	
1	9/3/2016	1	A_1	
2	12/2/2016	2	C_1	
3	4/15/2017	3	D_1	
4	5/24/2017	1	B_2	
5	5/25/2017	4	B_2	
6	6/6/2017	2	B_1	
7	6/10/2017	4	A_2	
8	6/10/2017	3	C_1	
9	7/20/2017	1	A_1	
10	8/11/2017	2	B_1	

Coding Techniques and Best Practices

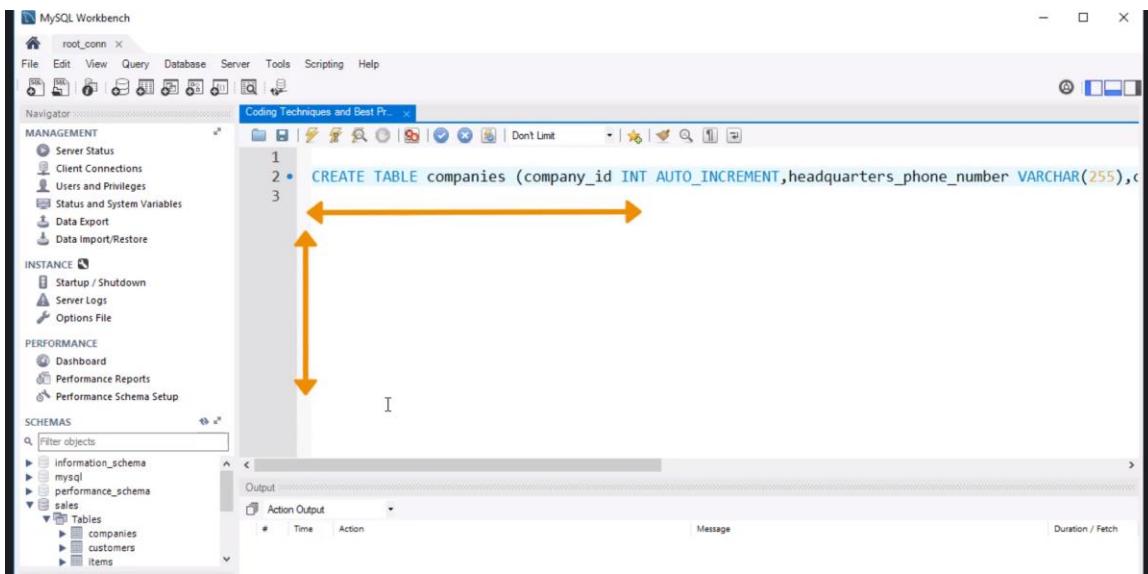


SQL

```
CREATE TABLE sales
(
    purchase_number INT,
    date_of_purchase DATE,
    customer_id VARCHAR(255),
    item_code VARCHAR(255),
    PRIMARY KEY (purcase_number)
);
```

Coding Techniques and Best Practices

Sales			
purchase_number	date_of_purchase	customer_id	item_code
1	9/3/2016	1	A_1
2	12/2/2016	2	C_1
3	4/15/2017	3	D_1
4	5/24/2017	1	B_2
5	5/25/2017	4	B_2
6	6/6/2017	2	B_1
7	6/10/2017	4	A_2
8	6/10/2017	3	C_1
9	7/20/2017	1	A_1
10	8/11/2017	2	B_1



MySQL Workbench screenshot showing a query editor with the following SQL code:

```
1 • use sales; I
2
3 • CREATE TABLE IF NOT EXISTS test (
4     numbers INT(10),
5     words VARCHAR(10)
6 );
```

The code is highlighted with syntax coloring. A large gray box labeled "Tab" is overlaid on the right side of the code area. The MySQL Workbench interface includes a Navigator pane on the left, an Output pane at the bottom, and a toolbar at the top.

MySQL Workbench screenshot showing the same SQL code as the first image. A red circle highlights the "Tab" key icon on the toolbar. Red handwritten notes are present on the right side of the screen, near the "Tab" label. The MySQL Workbench interface is identical to the first screenshot.

Section 9

AND

- = equal operator

in SQL, there are many other *Linking keywords and symbols*, called operators, that you can use with the WHERE clause

- AND	- EXISTS	- NOT EXISTS
- OR	- IS NULL	- IS NOT NULL
- IN	- NOT IN	- comparison operators
- LIKE	- NOT LIKE	- etc.
- BETWEEN... AND...		

Start from 96