

# Information Management System Lab

ECSE211L

**Bennett University**

## About Evaluation

- Continuous Lab Evaluation (10)
- Lab Examination (10)

# Data and Information

- Data is raw, unorganized facts that need to be processed.
- Data can be something simple, random and useless until it is organized.
  - Each student's exam marks
- When data is processed, organized, structured or presented in a given context to make it useful, it is called information.
  - The average score of a department or of the entire college is information that is derived from the given data.

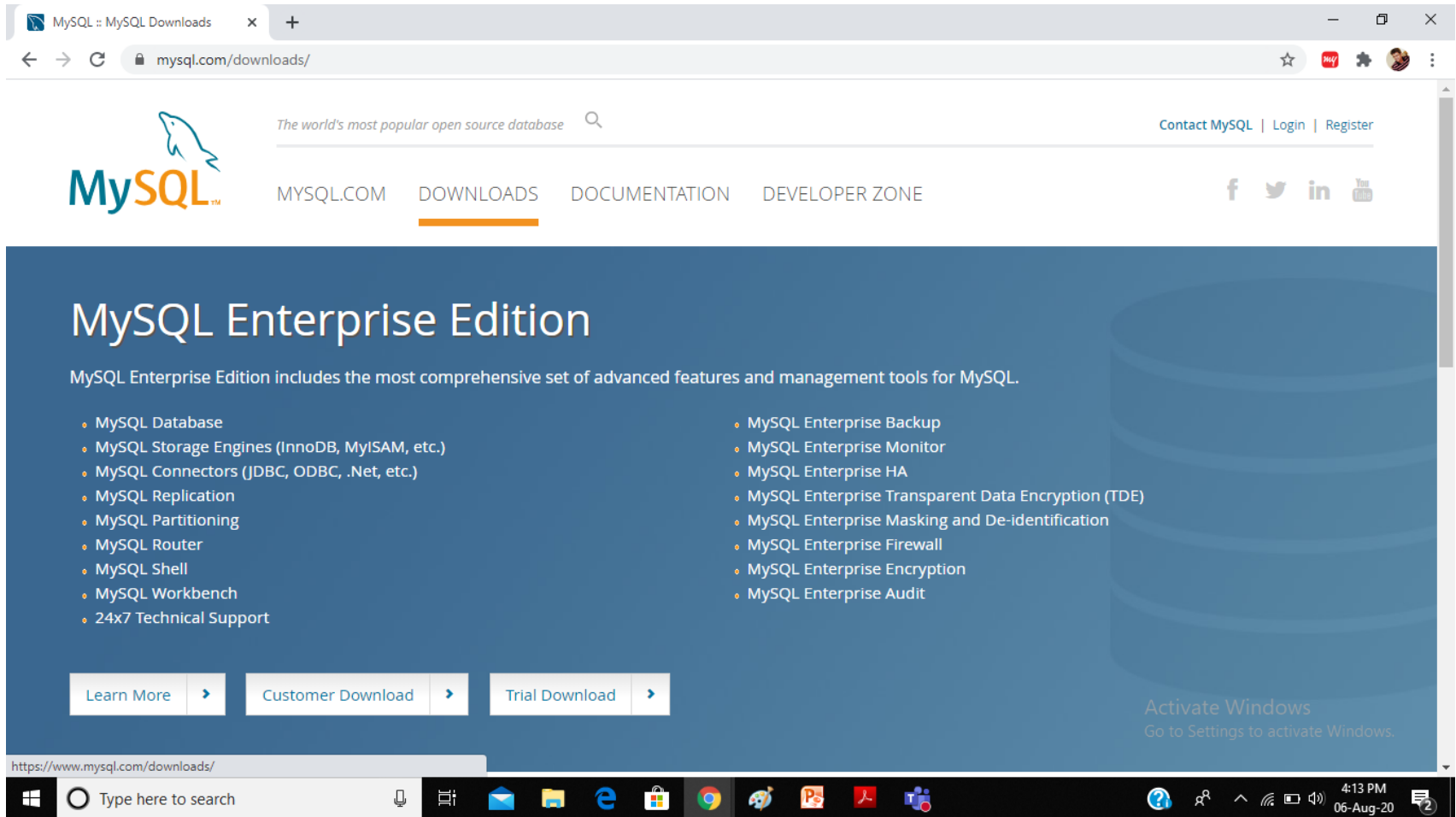
# Information Management System

- An Information Management System consists of
  - A collection of interrelated and persistent data (usually referred as database).
  - A set of application programs used to access, update and manage that data (usually referred as management system).

# Query Language

- SQL (structured query language) is a computer language aimed to store, manipulate, and retrieve data stored in relational databases.
- MySQL is one of the most popular open source SQL database management system.
- It is developed, distributed and supported by Oracle corporation.
- Supports including Windows, Linux, UNIX, Mac...

# Install MySQL



The screenshot shows a web browser window with the URL `mysql.com/downloads/`. The page features the MySQL logo and navigation links for `MYSQL.COM`, `DOWNLOADS` (highlighted), `DOCUMENTATION`, and `DEVELOPER ZONE`. The main heading is `MySQL Enterprise Edition`, followed by a description: "MySQL Enterprise Edition includes the most comprehensive set of advanced features and management tools for MySQL." Below this, there are two columns of features, each with a list of bullet points. At the bottom, there are three buttons: `Learn More`, `Customer Download`, and `Trial Download`. The Windows taskbar is visible at the bottom of the screen.

MySQL :: MySQL Downloads

mysql.com/downloads/

The world's most popular open source database

Contact MySQL | Login | Register

MySQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

## MySQL Enterprise Edition

MySQL Enterprise Edition includes the most comprehensive set of advanced features and management tools for MySQL.

- MySQL Database
- MySQL Storage Engines (InnoDB, MyISAM, etc.)
- MySQL Connectors (JDBC, ODBC, .Net, etc.)
- MySQL Replication
- MySQL Partitioning
- MySQL Router
- MySQL Shell
- MySQL Workbench
- 24x7 Technical Support

- MySQL Enterprise Backup
- MySQL Enterprise Monitor
- MySQL Enterprise HA
- MySQL Enterprise Transparent Data Encryption (TDE)
- MySQL Enterprise Masking and De-identification
- MySQL Enterprise Firewall
- MySQL Enterprise Encryption
- MySQL Enterprise Audit

Learn More > Customer Download > Trial Download >

Activate Windows  
Go to Settings to activate Windows.

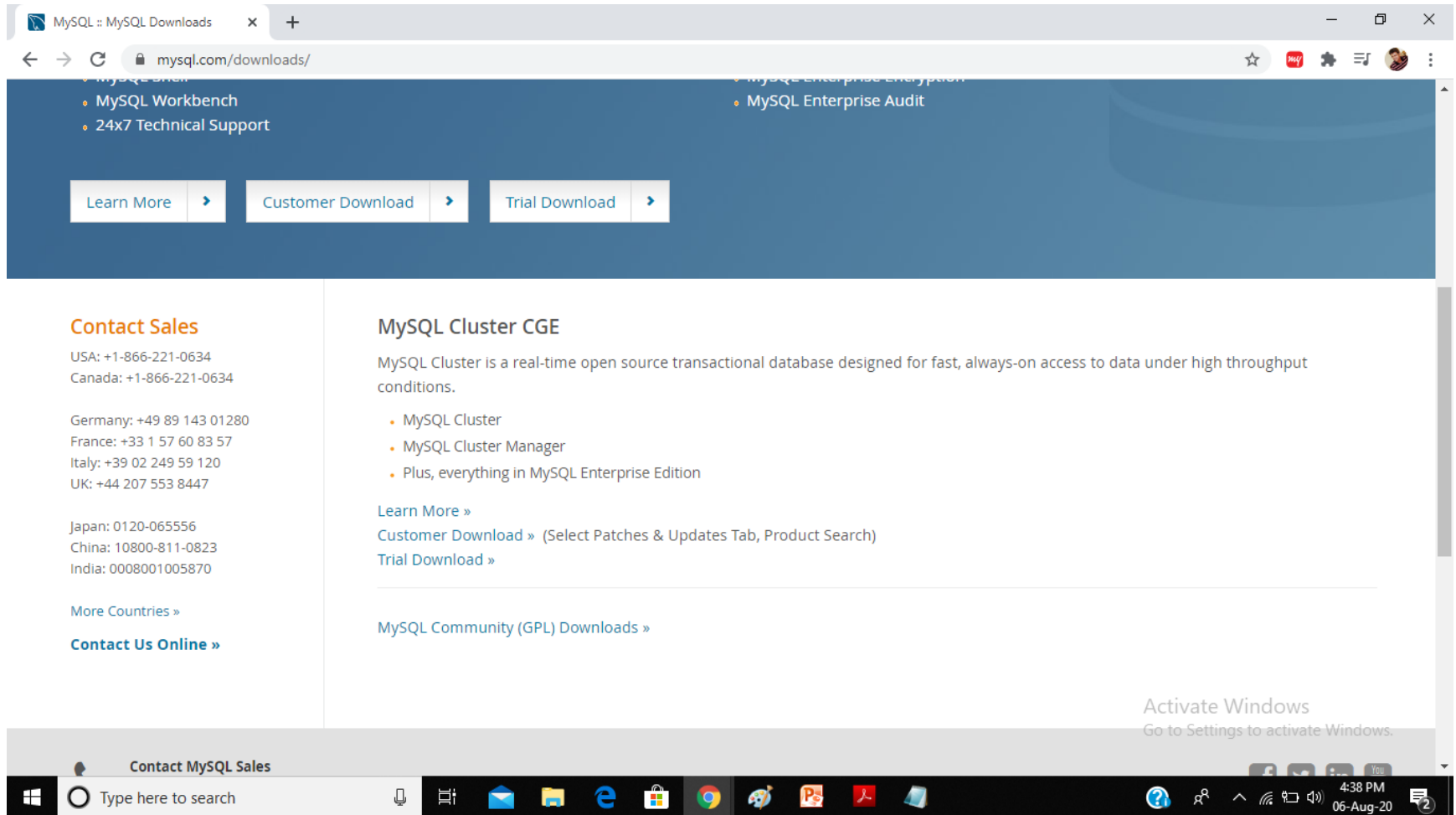
https://www.mysql.com/downloads/

Type here to search

4:13 PM  
06-Aug-20

Scroll down this page

# Install MySQL



The screenshot shows a web browser window with the address bar displaying "mysql.com/downloads/". The page has a dark blue header with navigation links: "MySQL Workbench", "24x7 Technical Support", "MySQL Enterprise Encryption", and "MySQL Enterprise Audit". Below the header are three buttons: "Learn More", "Customer Download", and "Trial Download".

**Contact Sales**

USA: +1-866-221-0634  
Canada: +1-866-221-0634

Germany: +49 89 143 01280  
France: +33 1 57 60 83 57  
Italy: +39 02 249 59 120  
UK: +44 207 553 8447

Japan: 0120-065556  
China: 10800-811-0823  
India: 0008001005870

[More Countries »](#)

[Contact Us Online »](#)

**MySQL Cluster CGE**

MySQL Cluster is a real-time open source transactional database designed for fast, always-on access to data under high throughput conditions.

- MySQL Cluster
- MySQL Cluster Manager
- Plus, everything in MySQL Enterprise Edition

[Learn More »](#)  
[Customer Download »](#) (Select Patches & Updates Tab, Product Search)  
[Trial Download »](#)

[MySQL Community \(GPL\) Downloads »](#)

Activate Windows  
Go to Settings to activate Windows.

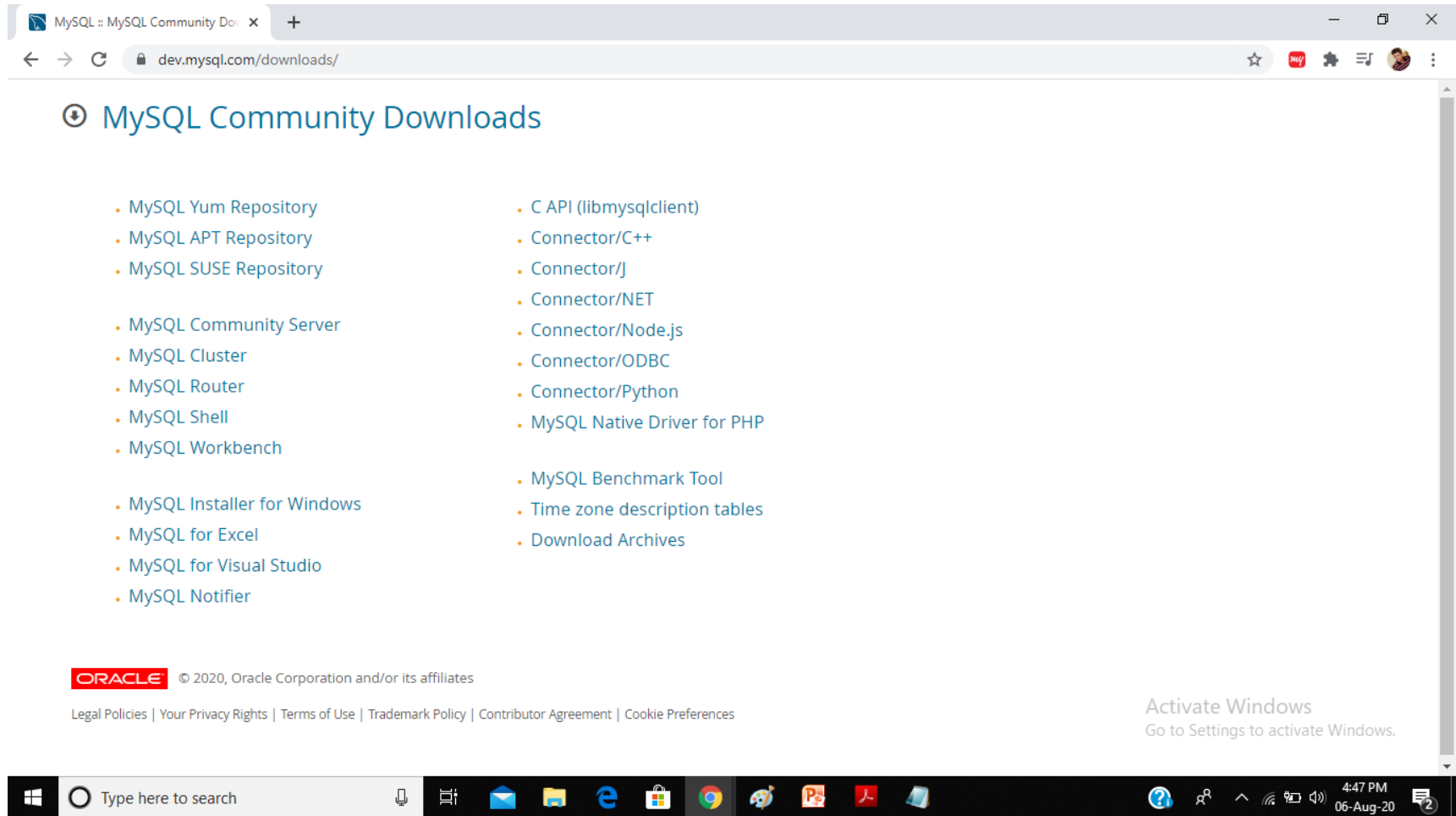
Contact MySQL Sales

Type here to search

4:38 PM  
06-Aug-20

Click on MySQL Community downloads

# Install MySQL



The screenshot shows a web browser window with the address bar displaying "dev.mysql.com/downloads/". The page title is "MySQL Community Downloads". The content is organized into two columns of links, each preceded by a small orange dot. The left column includes links for Yum, APT, and SUSE repositories; MySQL Community Server, Cluster, Router, Shell, and Workbench; and MySQL Installer for Windows, for Excel, for Visual Studio, and Notifier. The right column includes links for C API, Connector/C++, Connector/J, Connector/NET, Connector/Node.js, Connector/ODBC, Connector/Python, and MySQL Native Driver for PHP. A second set of links for MySQL Benchmark Tool, Time zone description tables, and Download Archives is positioned below the first column. At the bottom of the page, there is an Oracle logo, copyright information for 2020, and a list of legal policies. A Windows taskbar is visible at the very bottom, and a Windows activation watermark is present in the bottom right corner of the browser window.

MySQL :: MySQL Community Downloads

- [MySQL Yum Repository](#)
- [MySQL APT Repository](#)
- [MySQL SUSE Repository](#)
- [MySQL Community Server](#)
- [MySQL Cluster](#)
- [MySQL Router](#)
- [MySQL Shell](#)
- [MySQL Workbench](#)
- [MySQL Installer for Windows](#)
- [MySQL for Excel](#)
- [MySQL for Visual Studio](#)
- [MySQL Notifier](#)
- [C API \(libmysqlclient\)](#)
- [Connector/C++](#)
- [Connector/J](#)
- [Connector/NET](#)
- [Connector/Node.js](#)
- [Connector/ODBC](#)
- [Connector/Python](#)
- [MySQL Native Driver for PHP](#)
- [MySQL Benchmark Tool](#)
- [Time zone description tables](#)
- [Download Archives](#)

**ORACLE** © 2020, Oracle Corporation and/or its affiliates  
Legal Policies | Your Privacy Rights | Terms of Use | Trademark Policy | Contributor Agreement | Cookie Preferences

Activate Windows  
Go to Settings to activate Windows.

Click on the option according to your environment



# Install MySQL

The screenshot shows the MySQL download page for Windows. The browser address bar shows 'dev.mysql.com/downloads/installer/'. The page has tabs for 'General Availability (GA) Releases', 'Archives', and a help icon. The main heading is 'MySQL Installer 8.0.21'. Below it, a dropdown menu shows 'Microsoft Windows' selected. A button 'Looking for previous GA versions?' is next to it. There are two download options for Windows (x86, 32-bit), MSI Installer. The first option is 8.0.21, 24.5M, with a 'Download' button and MD5 checksum 'c f2b46ba35a4443f41fb8e94a0e91d93'. The second option is 8.0.21, 427.6M, with a 'Download' button and MD5 checksum 'b52294aa854356c266e9a9aec737ba08'. A note at the bottom suggests using MD5 checksums and GnuPG signatures to verify the integrity of the packages. The footer includes the Oracle logo, copyright notice '© 2020, Oracle Corporation and/or its affiliates', and links for 'Legal Policies', 'Your Privacy Rights', 'Terms of Use', 'Trademark Policy', 'Contributor Agreement', and 'Cookie Preferences'. An 'Activate Windows' watermark is visible in the bottom right corner.

MySQL :: Download MySQL Insta x +

dev.mysql.com/downloads/installer/

General Availability (GA) Releases Archives

## MySQL Installer 8.0.21

Select Operating System:  
Microsoft Windows

Looking for previous GA versions?

<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-web-community-8.0.21.0.msi)	8.0.21	24.5M	<a href="#">Download</a>
MD5: c f2b46ba35a4443f41fb8e94a0e91d93   <a href="#">Signature</a>			
<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-community-8.0.21.0.msi)	8.0.21	427.6M	<a href="#">Download</a>
MD5: b52294aa854356c266e9a9aec737ba08   <a href="#">Signature</a>			

We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

**ORACLE** © 2020, Oracle Corporation and/or its affiliates

[Legal Policies](#) | [Your Privacy Rights](#) | [Terms of Use](#) | [Trademark Policy](#) | [Contributor Agreement](#) | [Cookie Preferences](#)

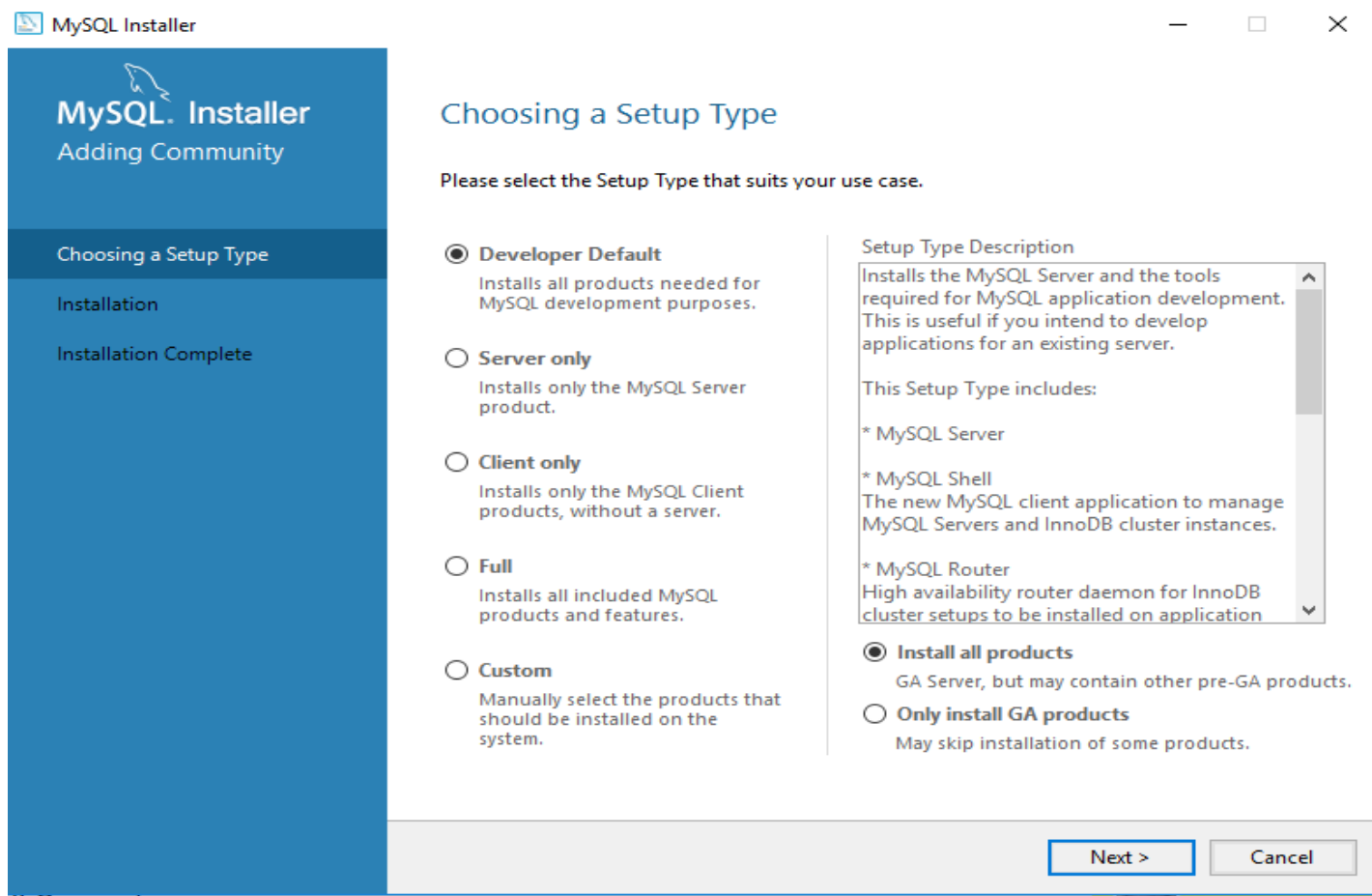
Activate Windows  
Go to Settings to activate Windows.

Type here to search

4:49 PM  
06-Aug-20

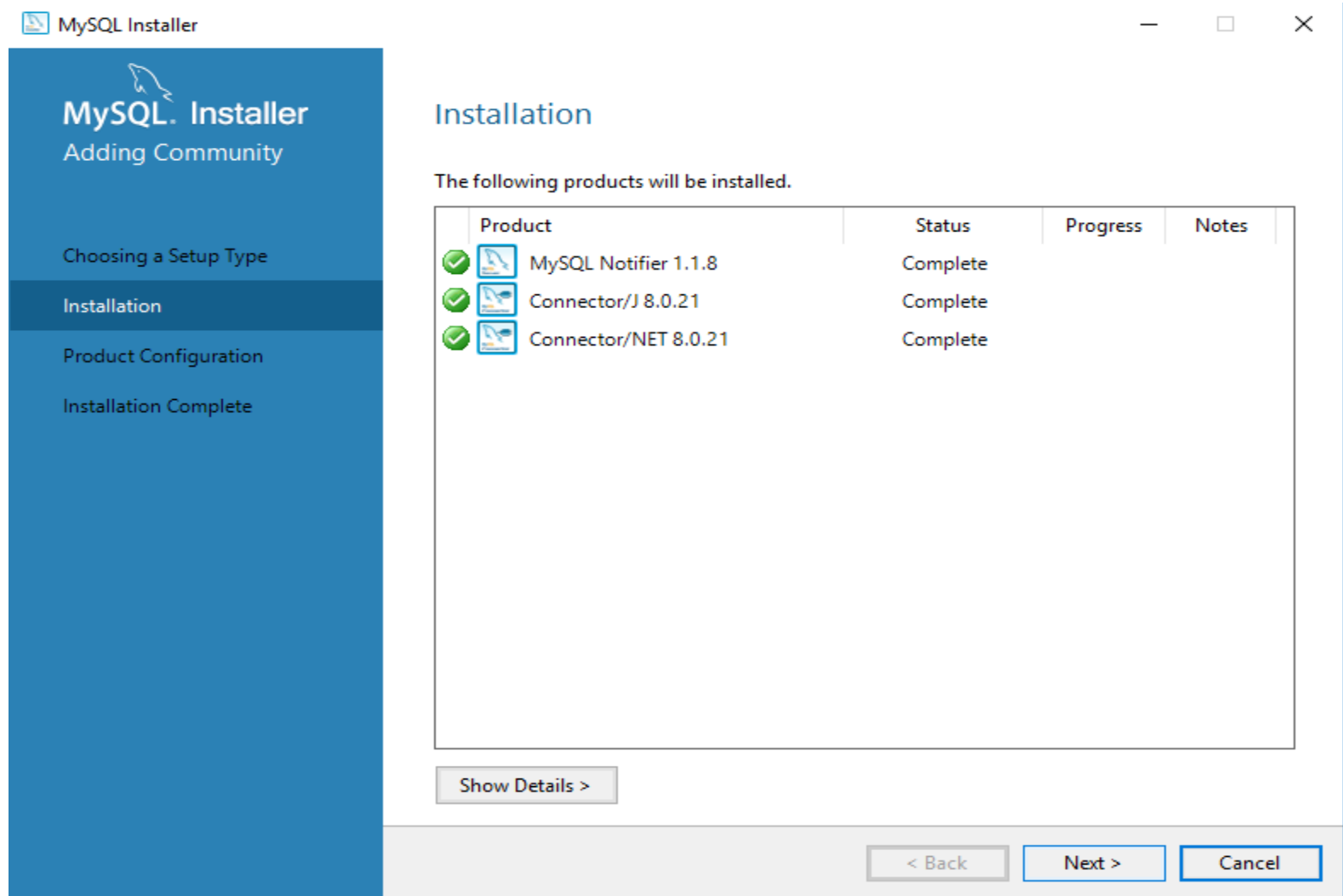
For windows, download from the 1<sup>st</sup> option

# MySQL Installer



Choose Developer and click next and continue the process

# MySQL Installer



click next and continue the process

# SQL categorization

SQL commands are mainly categorized into four categories as:

- DDL – Data Definition Language
- DML – Data Manipulation Language
- DQL – Data Query Language
- DCL – Data Control Language

# DDL

**DDL(Data Definition Language)** : Data Definition Language consists of the SQL commands that can be used to define the database schema.

- It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

**Some examples of DDL commands:**

- **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- **DROP** – is used to delete objects from the database.
- **ALTER**-is used to alter the structure of the database.
- **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT** –is used to add comments to the data dictionary.
- **RENAME** –is used to rename an object existing in the database.

# DML

- **DML(Data Manipulation Language)** : The SQL commands that deals with the manipulation of data present in the database belong to Data Manipulation Language.

## Examples of DML:

- **INSERT**— is used to insert data into a table.
- **UPDATE**— is used to update existing data within a table.
- **DELETE** — is used to delete records from a database table.

# DQL

- **DQL (Data Query Language) :**

The purpose of DQL Command is to get some schema relation based on the query.

## **Example of DQL:**

- **SELECT**– is used to retrieve data from the a database.

# DCL

- **DCL(Data Control Language)** : DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

## **Examples of DCL commands:**

- **GRANT**-gives user's access privileges to database.
- **REVOKE**-withdraw user's access privileges given by using the GRANT command.



# TCL

- **TCL(transaction Control Language)** : TCL commands deals with the transaction within the database.

## Examples of TCL commands:

- **COMMIT**– commits a Transaction.
- **ROLLBACK**– rollbacks a transaction in case of any error occurs.
- **SAVEPOINT**–sets a savepoint within a transaction.
- **SET TRANSACTION**–specify characteristics for the transaction.

# Create Database

- The CREATE DATABASE statement is used to create a new SQL database.

Syntax

- `CREATE DATABASE databasename;`

Example:

- `CREATE DATABASE testDB;`

To check whether the database is present, use following SQL command

- `SHOW DATABASES;`

# Drop Database

- The DROP DATABASE statement is used to drop an existing SQL database.

## Syntax

- **DROP DATABASE** *databasename*;

**Note:** Deleting a database will result in loss of complete information stored in the database!

## Example

- **DROP DATABASE** testDB;

**Tip:** You can check it in the list of databases with the following SQL command:  
SHOW DATABASES;

# Use Database

To use the database, we need to write following SQL command:

Syntax

- **USE** *databasename*;

Example

- **USE** testDB;

# Creating Table

- The CREATE TABLE statement is used to create a new table in a database.

## Syntax

- **CREATE TABLE** *table\_name* (  
    *column1 datatype*,  
    *column2 datatype*,  
    *column3 datatype*,  
    ....  
);
- The column parameters specify the names of the columns of the table.
- The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

# Create Table example

## SQL CREATE TABLE Example

- The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

### Example

- `CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);`

# Popular Data Types

Data Type Syntax	Explanation
CHAR( <i>size</i> )	Maximum size of 255 characters. size is the number of characters to store.
VARCHAR( <i>size</i> )	Maximum size of 255 characters. Variable-length string.
TEXT( <i>size</i> )	Maximum size of 65,535 characters. Where <b><i>size</i></b> is the number of characters to store.
BINARY( <i>size</i> )	Maximum size of 255 characters. Where <b><i>size</i></b> is the number of binary characters to store. Fixed-length strings.
INT( <i>m</i> )	Standard integer value. Signed values range from -2147483648 to 2147483647. Unsigned values range from 0 to 4294967295.
FLOAT( <i>m,d</i> )	Single precision floating point number. Where <b><i>m</i></b> is the total digits and <b><i>d</i></b> is the number of digits after the decimal.
DOUBLE( <i>m,d</i> )	Double precision floating point number. Where <b><i>m</i></b> is the total digits and <b><i>d</i></b> is the number of digits after the decimal.
DATE	Values range from '1000-01-01' to '9999-12-31'. Displayed as 'YYYY-MM-DD'.
DATETIME	Values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Displayed as 'YYYY-MM-DD HH:MM:SS'.

# Drop Table

- The DROP TABLE statement is used to drop an existing table in a database.

## Syntax

- **DROP TABLE** *table\_name*;
- **Note:** Deleting a table will result in loss of complete information stored in the table!

## Example:

- **DROP TABLE** team;



# Truncate Table

- The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

## Syntax

- `TRUNCATE TABLE table_name;`

## Example

- `TRUNCATE TABLE team;`

# Alter Table

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
- The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

# ALTER Table - ADD Column

To add a column in a table, use the following syntax:

- **ALTER TABLE** *table\_name*  
**ADD** *column\_name datatype*;

The following SQL adds an "Email" column to the "Persons" table:

Example

- **ALTER TABLE** Persons  
**ADD** Email varchar(255);

# ALTER Table - DROP Column

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

- `ALTER TABLE table_name  
DROP COLUMN column_name;`
- The following SQL deletes the "Email" column from the "Persons" table:

Example

- `ALTER TABLE Persons  
DROP COLUMN Email;`

# ALTER Table - ALTER/MODIFY Column

- To change the data type of a column in a table, use the following syntax:
- `ALTER TABLE table_name  
MODIFY COLUMN column_name datatype;`

Example

- `ALTER TABLE Persons  
ADD DateOfBirth date;`
- `ALTER TABLE Persons  
MODIFY COLUMN DateOfBirth year;`

# RENAME Table

- **RENAME TABLE** renames one or more tables. You must have ALTER and DROP privileges for the original table, and CREATE and INSERT privileges for the new table.

Syntax:

- **RENAME TABLE** tbname **TO** new\_tbname [, tbname2 **TO** new\_tbname2] ... ;

Example:

To rename a table named Persons to Employee, use this statement:

- **RENAME TABLE** Persons **TO** Employee;

We can also use

- **ALTER TABLE** Persons **RENAME** Employee;

# RENAME Contd.

- **RENAME TABLE**, unlike **ALTER TABLE**, can rename multiple tables within a single statement:
- **RENAME TABLE** old\_table1 **TO** new\_table1,  
old\_table2 **TO** new\_table2,  
old\_table3 **TO** new\_table3;
- Renaming operations are performed left to right. Thus, to swap two table names, do this (assuming that a table with the intermediary name tmp\_table does not already exist):
- **RENAME TABLE** old\_table **TO** tmp\_table,  
new\_table **TO** old\_table,  
tmp\_table **TO** new\_table;

# Add Comment

Add comment corresponding to a column

- ALTER TABLE Example

```
MODIFY COLUMN `id` int(10) COMMENT 'Look, I am a comment!';
```



# SELECT Command

The SELECT statement is used to select data from a database.

- The data returned is stored in a result table, called the result-set.

## Syntax

- **SELECT** *column1, column2, ...*  
**FROM** *table\_name*;
- Here, *column1, column2, ...* are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:
- **SELECT \* FROM** *table\_name*

# SELECT DISTINCT Command

- The SELECT DISTINCT statement is used to return only distinct (different) values.
- Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

## Syntax

- `SELECT DISTINCT column1, column2, ...  
FROM table_name;`

## Example

- `SELECT DISTINCT City FROM Persons;`

# WHERE Clause

- The WHERE clause is used to filter records.
- The WHERE clause is used to extract only those records that fulfill a specified condition.

## Syntax

- `SELECT column1, column2, ...`  
`FROM table_name`  
`WHERE condition;`

## Example

- `SELECT * FROM Persons`  
`WHERE City='Delhi';`

# MySQL Constraints

- **NOT NULL:** By default, a column can hold **NULL** values. The **NOT NULL** constraint enforces a column to **NOT** accept **NULL** values.
  - Eg. Select position, pname, role from team where shirt\_no is NOT NULL.
- **UNIQUE:** Ensures that all the values in columns are unique. (Eg. Roll no, Date of Birth, tshirt no of same team players etc.)
- **PRIMARY KEY:** Combination of NOT NULL and UNIQUE.
- **FOREIGN KEY:** Uniquely identify a row/record in any other database table.
- **CHECK:** The check constraints ensures that all values in a column specify certain condition.

# Example: Create Table

```
create table team(  
  position varchar(10),  
  pname varchar(30),  
  role varchar(20));
```

# Insert into Table

```
insert into team  
values ('P1','V. Kohli','Bat'),  
('P2','R. Sharma','Bat' ),  
('P3','M.Dhoni','Wicket'),  
('P4','J.Bumrah','Ball'),  
('P5','K.Yadav','Ball');
```

# How to display

- `SELECT position, pname from team;`
- `select * from team;`

P1	V. Kohli	Bat
P2	R. Sharma	Bat
P3	M.Dhoni	Wicket
P4	J.Bumrah	Ball
P5	K.Yadav	Ball

# Delete a Member

```
delete from team  
where position='P5';
```

```
select * from team;
```

P1	V. Kohli	Bat
P2	R. Sharma	Bat
P3	M.Dhoni	Wicket
P4	J.Bumrah	ball

```
Delete from team; //all the records will be deleted
```



# Update

update team

set pname='B.Kumar' where position='P4';

select \* from team;

P1	V. Kohli	Bat
P2	R. Sharma	Bat
P3	M.Dhoni	Wicket
P4	B.Kumar	Ball

# Alter Table

Alter table team

add (shirt\_no INT);

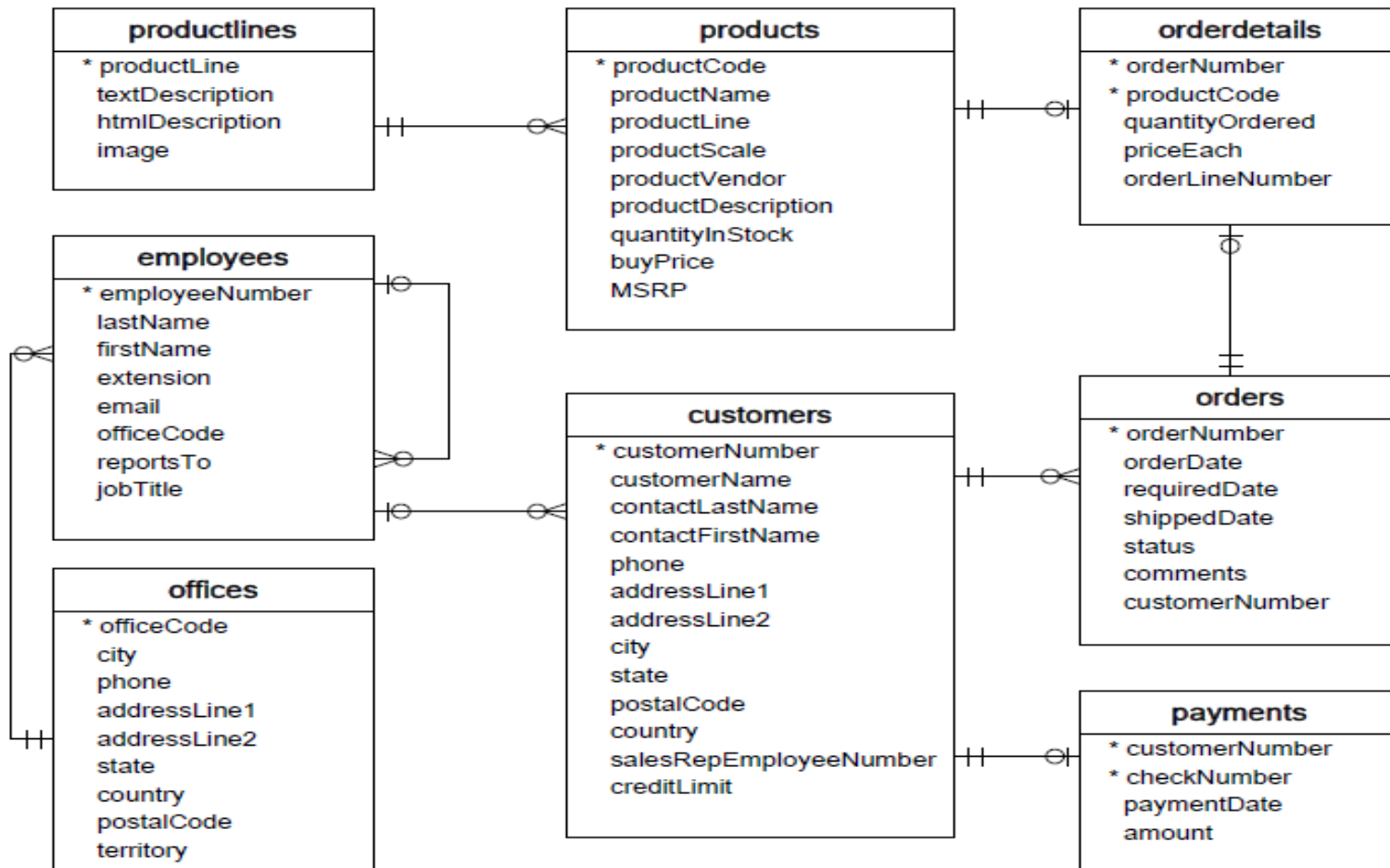
update team

set shirt\_no=18 where position='P1';

select \* from team;

P1	V. Kohli	Bat	18
P2	R. Sharma	Bat	NULL
P3	M.Dhoni	Wicket	NULL
P4	B.Kumar	Ball	NULL

# Use Case 1 (Shopping Record)



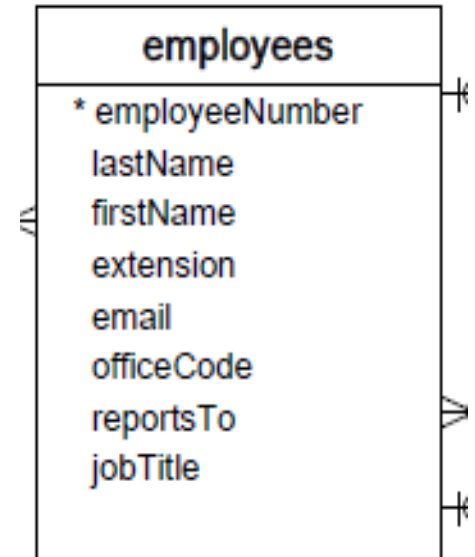
# Customers Table

```
CREATE TABLE customers (  
  customerNumber varchar(11) NOT NULL,  
  customerName varchar(50) NOT NULL,  
  contactLastName varchar(50) NOT NULL,  
  contactFirstName varchar(50) NOT NULL,  
  phone varchar(50) NOT NULL,  
  addressLine1 varchar(50) NOT NULL,  
  addressLine2 varchar(50) DEFAULT NULL,  
  city varchar(50) NOT NULL,  
  state varchar(50) DEFAULT NULL,  
  postalCode varchar(15) DEFAULT NULL,  
  country varchar(50) NOT NULL,  
  salesRepEmployeeNumber int(11) DEFAULT NULL,  
  creditLimit decimal(10,2) DEFAULT NULL,  
  PRIMARY KEY (customerNumber)) ;
```

customers	
*	customerNumber
	customerName
	contactLastName
	contactFirstName
	phone
	addressLine1
	addressLine2
	city
	state
	postalCode
	country
	salesRepEmployeeNumber
	creditLimit

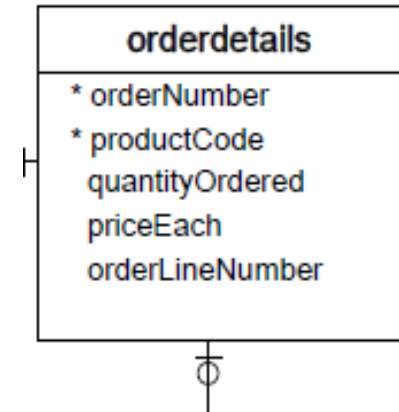
# Employees Table

```
CREATE TABLE employees (  
  employeeNumber int(11) NOT NULL,  
  lastName varchar(50) NOT NULL,  
  firstName varchar(50) NOT NULL,  
  extension varchar(10) NOT NULL,  
  email varchar(100) NOT NULL,  
  officeCode varchar(10) NOT NULL,  
  reportsTo int(11) DEFAULT NULL,  
  jobTitle varchar(50) NOT NULL,  
  PRIMARY KEY (employeeNumber));
```



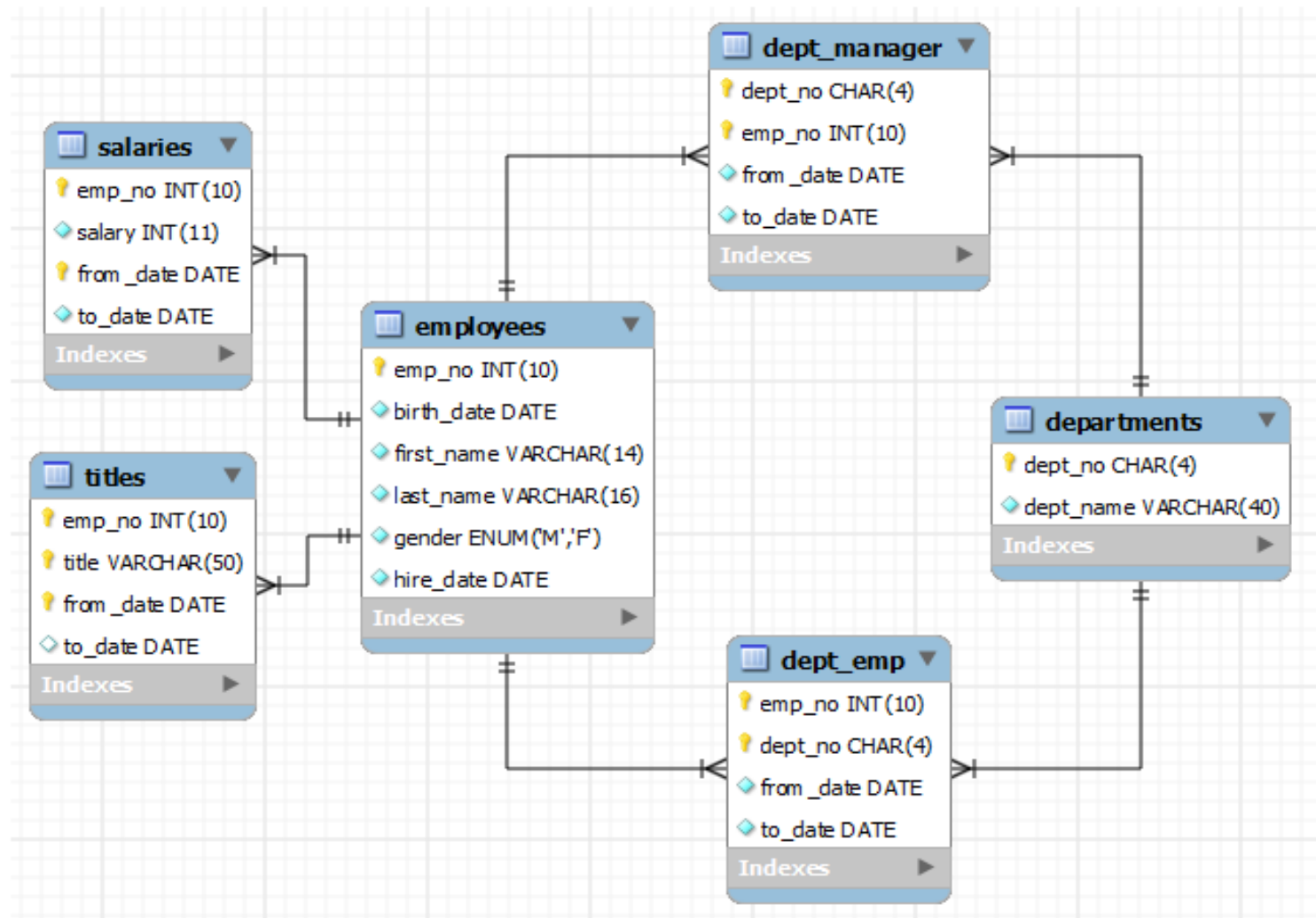
# Order Details Table

```
CREATE TABLE orderdetails (  
  orderNumber int(11) NOT NULL,  
  productCode varchar(15) NOT NULL,  
  quantityOrdered int(11) NOT NULL,  
  priceEach decimal(10,2) NOT NULL,  
  orderLineNumber smallint(6) NOT NULL,  
  PRIMARY KEY (orderNumber,productCode)) ;
```



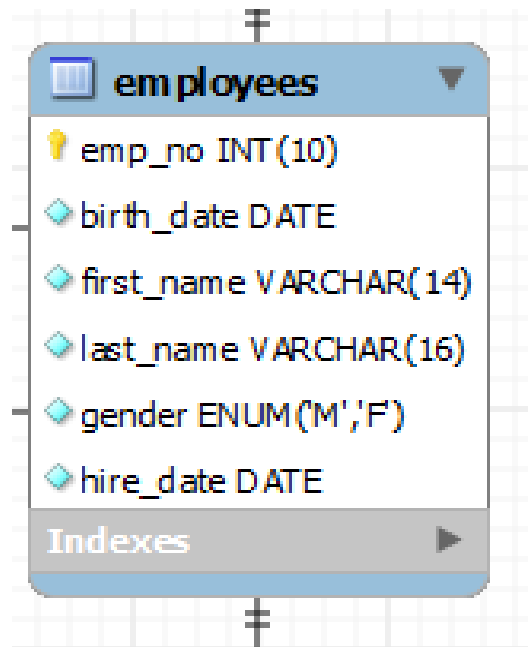
Other tables can be created in the similar process.

# Use Case 2: Office Record



# Employee Table

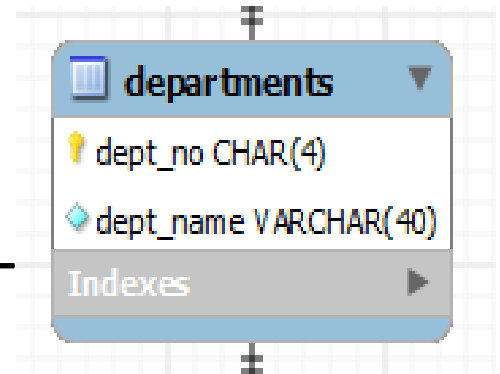
```
CREATE TABLE employees (  
    emp_no    INT          NOT NULL,  
    birth_date DATE        NOT NULL,  
    first_name VARCHAR(14) NOT NULL,  
    last_name  VARCHAR(16) NOT NULL,  
    gender     ENUM ('M','F') NOT NULL,  
    hire_date  DATE        NOT NULL,  
    PRIMARY KEY (emp_no)  
);
```





# Department Table

```
CREATE TABLE departments (  
    dept_no    CHAR(4)    NOT NULL,  
    dept_name  VARCHAR(40) NOT NULL  
    PRIMARY KEY (dept_no),  
    UNIQUE KEY (dept_name)  
);
```



Other tables can be created in the similar process.

# Assignment

Create following two tables:

Branch_name	Loan_number	amount
Downtown	L-170	3000
Redwood	L-230	4000
Perryridge	L-260	1700

Customer_name	Loan_number
Jones	L-170
Smith	L-230
Hayes	L-155

- 1) Find out the Branch\_name where amount  $\geq 2000$ .
- 2) Find out the name of the customer whose loan\_number is L-170.
- 3) Delete the customer whose loan\_number is 155.
- 4) Add gender column in the second table.