PHP: 5.2, MYSQL: 5.1 , APACHE : 2.2

MYSQL:-

1.how can find second highest salary ?

select salary from employees e where 1=(select count(\*) from employees where e.salary<salary )

SELECT MAX(SALARY) FROM tbl\_name WHERE SALARY < (SELECT MAX(SALARY) FROM tbl\_name)

2.Differences between MYISAM and InnoDB ENGINE:-

MYISAM does not support the foreign key constraint and

transaction but InnoDB support it.

MYISAM is faster then the InnoDB but in case of perforing

the count operation it takes more time then the InnoDB.

MYISAM occupies less memory sapce for tables rather than

InnoDB tables

**InnoDB** has row-level locking, **MyISAM** can only do full table-level locking

As InnoDB supports row-level locking which means inserting and updating is much faster as compared with MyISAM.

InnoDB does not support FULLTEXT index while MyISAM supports.

3.

Groupby: It is a way to sub-total your results,or perform

some other 'aggregate' functions on them

ex:-

SELECT student\_name, AVG(test\_score)

FROM student

GROUP BY student\_name

orderby: order by use for sorting the result,like ascending or deciding.

PHP :-

**How to find duplicate email records in users table?**

SELECT u1.first\_name, u1.last\_name, u1.email FROM users as u1

INNER JOIN (

    SELECT email FROM users GROUP BY email HAVING count(id) > 1

    ) u2 ON u1.email = u2.email;

4. Joins

The INNER JOIN keyword return rows when there is at least one match in both tables. If there are rows in "Persons" that do not have matches in "Orders", those rows will NOT be listed.

The LEFT JOIN keyword returns all rows from the left table (table\_name1), even if there are no matches in the right table (table\_name2).

The RIGHT JOIN keyword returns all the rows from the right table (table\_name2), even if there are no matches in the left table (table\_name1).

The FULL JOIN keyword return rows when there is a match in one of the tables.

5.having is used when the condition has to be applied on the result of a group by clause.

“i”. select current\_date() + INTERVAL 1 DAY AS Tomorrow

“ii” select current\_date() - INTERVAL 1 DAY AS yesterday

https://mail.google.com/mail/u/0/images/cleardot.gif

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

6.Database triggers are SQL statements storing in the database catalog. Once a trigger is activated by database events such as UPDATE, DELETE or INSERT, it will execute either before or after the event that initiated it.

1.ob\_start:-

This function will turn output buffering on. While output buffering is active no output is sent from the script (other than headers), instead the output is stored in an internal buffer

2.ob\_flush:-

This function will send the contents of the output buffer (if any). If you want to further process the buffer's contents you have to call ob\_get\_contents() before ob\_flush() as the buffer contents are discarded after ob\_flush() is called.

3.Constructors:-

Constructors are functions in a class that are automatically called when you create a new instance of a class with new. A function becomes a constructor, when it has the same name as the class. If a class has no constructor, the constructor of the base class will be called, if it exists

<?php

class BaseClass {

function \_\_construct() {

echo "In BaseClass constructor\n<br>";

}

}

class SubClass extends BaseClass {

function \_\_construct() {

parent::\_\_construct();

echo "In SubClass constructor\n<br>";

}

function test()

{

echo "testingggggggggg<br>";

}

}

$obj= new SubClass();

$obj->test();

?>

4. Overloading

.The Methods with the same name but it differs by types of arguments and no, of arguments.

.Overloading is Static Polymorphism.

.Overloading refers to situation where there are two or more methods in a class with the same name but with different parameters list.

Overriding

.The Methods with the same name and same no,,of arguments and types,but one is in base class and second one is in derived class.

.Overriding is dynamic Polymorphism.

.Overriding refers to a situation where sub-class inherits a method from Base class which may result into adding or changing the method behavior

ex:-<?php

class Foo {

function myFoo() {

return "Foo";

}

}

class Bar extends Foo {

function myFoo() {

return "Bar";

}

}

$foo = new Foo;

$bar = new Bar;

echo($foo->myFoo()); //"Foo"

echo($bar->myFoo()); //"Bar"

5.explain public private protected in php

private - Only the current class will have access to the field or method.

protected - Only the current class and subclasses (and sometimes also same-package classes) of this class will have access to the field or method.

public - Any class can refer to the field or call the method

6.Interface

Interface is a class with a list of methods which must be created in classes where it is implemented.

An interface is declared with the interface keyword. It can contain constants and method declarations, but not method bodies

7.What is Normalization?

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals

There is 3 Normalization

1st form of Normalization is row wise Normalization

2nd form of Normalization is column wise Normalization

3rd form of Normalization is Indexing wise Normalization

8.serialize() Generates a storable representation of a value

This is useful for storing or passing PHP values around without losing their type and structure

1. <?php
2. $serialized\_data = serialize(**array**('Math', 'Language', 'Science'));
3. echo  $serialized\_data . '<br>';
4. ?>

Output :

1. a:3:{i:0;s:4:"Math";i:1;s:8:"Language";i:2;s:7:"Science";}

unserialize() can use this string to recreate the original variable values.

9.Explain about encapsulation?

Encapsulation passes the message without revealing the exact functional details of the class. It allows only the relevant information to the user without revealing the functional mechanism through which a particular class had functioned.

10.Explain about abstraction?

Abstraction simplifies a complex problem to a simpler problem by specifying and modeling the class to the relevant problem scenario. It simplifies the problem by giving the class its specific class of inheritance.

11.Explain the mechanism of composition?

Composition helps to simplify a complex problem into an easier problem. It makes different classes and objects to interact with each other thus making the problem to be solved automatically. It interacts with the problem by making different classes and objects to send a message to each other

12.Explain what a method is ?

A method will affect only a particular object to which it is specified. Methods are verbs meaning they define actions which a particular object will perform

13.PHP Design Patterns:

A design pattern is like a template which guides us to solve a problem. Design patterns are not finished design that can be transformed directly into code. In Object Oriented based design patterns we can get the interactions and relationships between classes or objects

**14) Explain about UML?**  
UML or unified modeling language is regarded to implement complete specifications and features of object oriented language. Abstract design can be implemented in object oriented programming languages. It lacks implementation of polymorphism on message arguments which is a OOPs feature

UML stands for Unifed Modeling Language.UML is used to manage large and complex systems.

With UML you can:

Manage project complexity.

create database schema.

Produce reports.

16.A friend function is a non member function of a class, that

is declared as a friend using the keyword "friend" inside the class. All the access permissions are given to the function.

**15A) Explain about object oriented databases?**  
Object oriented databases are very popular such as relational database management systems. Object oriented databases systems use specific structure through which they extract data and they combine the data for a specific output. These DBMS use object oriented languages to make the process easier

17.

Abstract class: abstract classes are the class where one or more

methods are abstract but not necessarily all method has to be abstract.

Abstract methods are the methods, which are declare in its class but not define. The definition of those methods must be in its extending class.

Interface: Interfaces are one type of class where all the methods are abstract. That means all the methods only declared but not defined. All the methods must be define by its implemented class.

Explain about a class?  
Class describes the nature of a particular thing. Structure and modularity is provided by a Class in object oriented programming environment. Characteristics of the class should be understandable by an ordinary non programmer and it should also convey the meaning of the problem statement to him. Class acts like a blue print.

5) Explain about message passing in object oriented programming?  
Message passing is a method by which an object sends data to another object or requests other object to invoke method. This is also known as interfacing. It acts like a messenger from one object to other object to convey specific instructions.

**6) Explain what is an object?**  
An object is a combination of messages and data. Objects can receive and send messages and use messages to interact with each other. The messages contain information that is to be passed to the recipient object.

**7) Explain what a method is?**  
A method will affect only a particular object to which it is specified. Methods are verbs meaning they define actions which a particular object will perform. It also defines various other characteristics of a particular object.

* **9) Parent class:** A class that is inherited from by another class. This is also called a base class or super class.
* **10) Child Class:** A class that inherits from another class. This is also called a subclass or derived class.

18. The split() function splits the string into an array using a regular expression and returns an array.

<?php print\_r(str\_split("Hello")); ?>

OUTPUT :-

Array

(

[0] => H

[1] => e

[2] => l

[3] => l

[4] => o

)

The explode() function splits the string by string.

<?php

$str = "Hello world. It's a beautiful day.";

print\_r (explode(" ",$str));

?>

OUTPUT:-

Array

(

[0] => Hello

[1] => world.

[2] => It's

[3] => a

[4] => beautiful

[5] => day.

)

**Explain what is an object?**  
An object is a combination of messages and data. Objects can receive and send messages and use messages to interact with each other. The messages contain information that is to be passed to the recipient object.

**Differences between abstract class and interface in PHP**

Following are some main difference between abstract classes and interface in php

1. In abstract classes this is not necessary that every method should be abstract. But in interface every method is abstract.
2. Multiple and multilevel both type of inheritance is possible in interface. But single and multilevel inheritance is possible in abstract classes.
3. Method of php interface must be public only. Method in abstract class in php could be public or protected both.
4. In abstract class you can define as well as declare methods. But in interface you can only defined your methods.

PHP OOPS IMP LINK

<http://www.techflirt.com/tutorials/oop-in-php/abstract-classes-interface.html>

19. Htaccess files provide a way to make configuration changes on a per-directory basis. A file, containing one or more configuration directives, is placed in a particular document directory, and the directives apply to that directory, and all subdirectories thereof

Or

.htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.

* [Redirects](http://www.htaccess-guide.com/redirects/)
* [Password protection](http://www.htaccess-guide.com/password-protection/)
* [Deny visitors by IP address](http://www.htaccess-guide.com/deny-visitors-by-ip-address/)
* [Setting server timezone](http://www.htaccess-guide.com/setting-server-timezone/)

**Polymorphism :**

* Polymorphism is the process of creating several objects from specific base classes.
* A separate object can have separate properties and methods which work separately to other objects.
* However a set of objects could be derived from a parent object and keep some properties of the parent class.
* There is two ways to achieve in polymorphism :
  1. Compile time(Overloading)
  2. Run time(Overriding)

#### Overloading

* If you create more than one function with same name but with different signature.
* That means it could have different number of parameters, different datatype of the parameter etc. that is known as Compile Time Polymorphism
* PHP does not support Overloading.

#### Overriding

* If you override any method (either declared as protected or public) declared in superclass and perform anything as you wish in subclass that is known as Run Time Polymorphism.

**PHP Polymorphism Function Example:**

<?php

classA{

functionDisp(){

echo"Inside the Base class<br/>";}

}

classB extends A{

functionDisp(){

echo"Inside the Chlid class<br/>";}}

classC extends A  
{

}

$obj=new B();

$obj->Disp();

$obj2=new C();

$obj2->Disp();

?>

**Output:**

Inside the Chlid class  
Inside the Base class

#### Final Method

##### **.**If you declare method as final in super class it can't be overridden in any of its subclass **Example :**

[view plaincopy to clipboardprint?](http://www.tkhts.com/php/final.jsp)

1. <?php
2. **class** Super
4. {
5. **public** final function display()
6. {
7. echo "Inside the final function";
8. }
9. }
10. **class** Sub extends Super
11. {
12. function display()
13. {
14. echo "Final function";
15. }
16. }
17. $obj=**new** Sub();
18. s$obj->display();
19. ?>

#### Class as Final

* If you can declare a class as final, which will intercept anyone from extending it.

##### **Example :**

[view plaincopy to clipboardprint?](http://www.tkhts.com/php/final.jsp)

1. <?php
2. final **class** Super
3. {
4. **public** function display()
5. {
6. echo "Inside the final function";
7. }
8. }
9. **class** Sub extends Super
10. {
11. function display()
12. {
13. echo "Inside the final function";
14. }
15. }
16. $obj=**new** Sub();
17. $obj->display();
18. ?>

##### **Output :**

Fatal error: Class Sub may not inherit from final class (Super)

##### **Output :**

Fatal error: Cannot override final method Super::display()

ARRAY FUNCTIONS :

1. [in\_array](http://www.php.net/manual/en/function.in-array.php)— Checks if a value exists in an array

<?php  
$os = array("Mac", "NT", "Irix", "Linux");  
if (in\_array("Irix", $os)) {  
    echo "Got Irix";  
}

?>

2.[array\_pop](http://www.php.net/manual/en/function.array-pop.php) — Pop the element off the end of array

<?php  
$stack = array("orange", "banana", "apple", "raspberry");  
$fruit = array\_pop($stack);  
print\_r($stack);

?>

OUT PUT :

Array

(

[0] => orange

[1] => banana

[2] => apple

)

3.[array\_push](http://www.php.net/manual/en/function.array-push.php) — Push one or more elements onto the end of array<?php  
$stack = array("orange", "banana");  
array\_push($stack, "apple", "raspberry");  
print\_r($stack);  
?>

OUTPUT

Array

(

[0] => orange

[1] => banana

[2] => apple

[3] => raspberry

)

[array\_unique](http://www.php.net/manual/en/function.array-unique.php) — Removes duplicate values from an array

<?php  
$input = array("a" => "green", "red", "b" => "green", "blue", "red");  
$result = array\_unique($input);  
print\_r($result);?>

[array\_unshift](http://www.php.net/manual/en/function.array-unshift.php) — Prepend one or more elements to the beginning of an array

[array\_values](http://www.php.net/manual/en/function.array-values.php) — Return all the values of an array

[array\_walk\_recursive](http://www.php.net/manual/en/function.array-walk-recursive.php) — Apply a user function recursively to every member of an array

[array\_walk](http://www.php.net/manual/en/function.array-walk.php) — Apply a user function to every member of an array

[array](http://www.php.net/manual/en/function.array.php) — Create an array

[arsort](http://www.php.net/manual/en/function.arsort.php) — Sort an array in reverse order and maintain index association

[asort](http://www.php.net/manual/en/function.asort.php) — Sort an array and maintain index association

[compact](http://www.php.net/manual/en/function.compact.php) — Create array containing variables and their values

[count](http://www.php.net/manual/en/function.count.php) — Count all elements in an array, or something in an object

[current](http://www.php.net/manual/en/function.current.php) — Return the current element in an array

[each](http://www.php.net/manual/en/function.each.php) — Return the current key and value pair from an array and advance the array cursor

[end](http://www.php.net/manual/en/function.end.php) — Set the internal pointer of an array to its last element

[extract](http://www.php.net/manual/en/function.extract.php) — Import variables into the current symbol table from an array

[in\_array](http://www.php.net/manual/en/function.in-array.php) — Checks if a value exists in an array

[key](http://www.php.net/manual/en/function.key.php) — Fetch a key from an array

[krsort](http://www.php.net/manual/en/function.krsort.php) — Sort an array by key in reverse order

[ksort](http://www.php.net/manual/en/function.ksort.php) — Sort an array by key

[list](http://www.php.net/manual/en/function.list.php) — Assign variables as if they were an array

[natcasesort](http://www.php.net/manual/en/function.natcasesort.php) — Sort an array using a case insensitive "natural order" algorithm

[natsort](http://www.php.net/manual/en/function.natsort.php) — Sort an array using a "natural order" algorithm

[next](http://www.php.net/manual/en/function.next.php) — Advance the internal array pointer of an array

[pos](http://www.php.net/manual/en/function.pos.php) — Alias of current

[prev](http://www.php.net/manual/en/function.prev.php) — Rewind the internal array pointer

[range](http://www.php.net/manual/en/function.range.php) — Create an array containing a range of elements

[reset](http://www.php.net/manual/en/function.reset.php) — Set the internal pointer of an array to its first element

[rsort](http://www.php.net/manual/en/function.rsort.php) — Sort an array in reverse order

[shuffle](http://www.php.net/manual/en/function.shuffle.php) — Shuffle an array

[sizeof](http://www.php.net/manual/en/function.sizeof.php) — Alias of count

[sort](http://www.php.net/manual/en/function.sort.php) — Sort an array

[uasort](http://www.php.net/manual/en/function.uasort.php) — Sort an array with a user-defined comparison function and maintain index association

[uksort](http://www.php.net/manual/en/function.uksort.php) — Sort an array by keys using a user-defined comparison function

[usort](http://www.php.net/manual/en/function.usort.php) — Sort an array by values using a user-defined comparison function

The var\_dump() function is used to displays structured information (type and value) about one or more variables.

<?php  
$people = array("Peter", "Joe", "Glenn", "Cleveland");  
  
echo current($people) . "<br>"; // The current element is Peter  
echo next($people) . "<br>"; // The next element of Peter is Joe  
echo current($people) . "<br>"; // Now the current element is Joe  
echo prev($people) . "<br>"; // The previous element of Joe is Peter  
echo end($people) . "<br>"; // The last element is Cleveland  
echo prev($people) . "<br>"; // The previous element of Cleveland is Glenn  
echo current($people) . "<br>"; // Now the current element is Glenn  
echo reset($people) . "<br>"; // Moves the internal pointer to the first element of the array, which is Peter  
echo next($people) . "<br>"; // The next element of Peter is Joe  
  
print\_r (each($people)); // Returns the key and value of the current element (now Joe), and moves the internal pointer forward  
?>

Reply-to: Reply-to is where to delivery the reply of the mail.  
  
Return-path: Return path is when there is a mail delivery failure occurs then where to delivery the failure notification.

[How we use copy() and move() file uploading functions in PHP?](http://r4r.co.in/answer.php?id=2447&option=PHP%20Subjective)

When we use copy function it copy the file from source location to destination location and keep original file to the source location. Syntax: copy($\_FILES['uploadedfile']['tmp\_name'], $destination\_path) Where as when we use move function it copy the file from source location to destination location and delete original file from source location. Syntax: move\_uploaded\_file($\_FILES['uploadedfile']['tmp\_name'], $destination\_path)

SESSION: A PHP session variable is used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.

Web Services: Web Services are a way to send and receive information between remote programs   
Types of web services :-  
XML-RPC :- XML-Remorte Procedure Call,   
SOAP:- Simple Object Access Protocol ,   
 REST,

Free-form.  
Web services urls:-  
<http://www.slideshare.net/mgirouard/creating-and-consuming-web-services-in-php-5>  
<http://davidwalsh.name/web-service-php-mysql-xml-json>

What is URL rewriting  
Using URL rewriting we can convert dynamic URl to static URL Static URLs are known to be better than Dynamic URLs because of a number of reasons 1. Static URLs typically Rank better in Search Engines. 2. Search Engines are known to index the content of dynamic pages a lot slower compared to static pages. 3. Static URLs are always more friendlier looking to the End Users. along with this we can use URL rewriting in adding variables [cookies] to the URL to handle the sessions.

**session\_set\_save\_handler()**sets the user-level session storage functions which are used for storing and retrieving data associated with a session. This is most useful when a storage method other than those supplied by PHP sessions is preferred. i.e. Storing the session data in a local database.

What is CURL ?  
CURL means Client URL Library  
curl is a command line tool for transferring files with URL syntax, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE. curl supports SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, user+password authentication (Basic, Digest, NTLM, Negotiate, kerberosâ€¦), file transfer resume, proxy tunneling and a busload of other useful tricks.  
CURL allows you to connect and communicate to many different types of servers with many different types of protocols. libcurl currently supports the http, https, ftp, gopher, telnet, dict, file, and ldap protocols. libcurl also supports HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading (this can also be done with PHPâ€™s ftp extension), HTTP form based upload, proxies, cookies, and user+password authentication.

**WORD PRESS QUS&ANS**

1.difference between widget and plugin wordpress:-

A plugin is a collection of files that either change the way WordPress functions, or adds/removes functionality.

Plugins are written (mostly) in the PHP-language and they are stored in wp-content/plugins directory. These plugins can be activated or deactivated through the WordPress admin panel by the blog admin.

- Widgets are placeholders or user interface elements that present data.

Sidebar widgets can be reordered, renamed, enabled or disabled to change the looks, feel and contents of the user interface. Some plugins present their own user interface as a sidebar widget. Sometimes plugins provide optional widgets (such as polls widget of wp-polls, or the language chooser widget of the Gengo plugin). Sometimes plugins have no widgets.

sudo chmod 777 /var/www

sudo /etc/init.d/apache2 restart

* [**MyISAM**](https://dev.mysql.com/doc/refman/5.1/en/myisam-storage-engine.html): The default MySQL storage engine and the one that is used the most in Web, data warehousing, and other application environments. **MyISAM** is supported in all MySQL configurations, and is the default storage engine unless you have configured MySQL to use a different one by default.
* [**InnoDB**](https://dev.mysql.com/doc/refman/5.1/en/innodb-storage-engine.html): A transaction-safe (ACID compliant) storage engine for MySQL that has commit, rollback, and crash-recovery capabilities to protect user data. **InnoDB** row-level locking (without escalation to coarser granularity locks) and Oracle-style consistent nonlocking reads increase multi-user concurrency and performance. **InnoDB**stores user data in clustered indexes to reduce I/O for common queries based on primary keys. To maintain data integrity, **InnoDB** also supports **FOREIGN KEY** referential-integrity constraints.
* [**Memory**](https://dev.mysql.com/doc/refman/5.1/en/memory-storage-engine.html): Stores all data in RAM for extremely fast access in environments that require quick lookups of reference and other like data. This engine was formerly known as the **HEAP** engine.
* [**Merge**](https://dev.mysql.com/doc/refman/5.1/en/merge-storage-engine.html): Enables a MySQL DBA or developer to logically group a series of identical **MyISAM** tables and reference them as one object. Good for VLDB environments such as data warehousing.
* [**Archive**](https://dev.mysql.com/doc/refman/5.1/en/archive-storage-engine.html): Provides the perfect solution for storing and retrieving large amounts of seldom-referenced historical, archived, or security audit information.
* [**Federated**](https://dev.mysql.com/doc/refman/5.1/en/federated-storage-engine.html): Offers the ability to link separate MySQL servers to create one logical database from many physical servers. Very good for distributed or data mart environments.
* [**NDB**](https://dev.mysql.com/doc/refman/5.1/en/mysql-cluster.html) (also known as [**NDBCLUSTER**](https://dev.mysql.com/doc/refman/5.1/en/mysql-cluster.html))—This clustered database engine is particularly suited for applications that require the highest possible degree of uptime and availability

**Foreign key :**

Foreign key is a column in one table which is primary key on another table. Foreign key and Primary key is used to define relationship between two tables in relational database

One more *difference between primary key and foreign key* is that **foreign key can be duplicate** opposite to primary key which is always unique.

**Delete Duplicate Records :**

DELETE FROM states

where sname in(

SELECT sname

FROM (

select sname

FROM states

GROUP BY sname having COUNT(\*)>2 )temp

);

**PHP4 vs PHP5 (Additional Features of PHP5)**

Constructors and Destructors

In PHP4, Constructor have same name as the Class name.

In PHP5, name Constructors as \_construct and Destructors as \_destruct().

Passed by References

In PHP4, everything was passed by value.

In PHP5, all objects are passed by references.

Abstract

In PHP5, we can declare a class as abstract.

Static Methods and Properties

In PHP5, Static Methods and Properties are also available. When you declare a class as static, then you can access using :: operator without creating an instances of class

\_autoload()

PHP5 introduces a special function called \_autoload().

Final

PHP5 allows you to declare a class or method as final.

Magic Methods

PHP5 introduces magic methods such as \_call, \_get, \_set and \_tostring

Visibility

PHP5 has 3 level of visibilities

Public: methods are accessible to everyone including objects outside the class

Private: methods are accessible to the class itself

Protected: methods are accessible to the class itself and inherited classes.

Exception

PHP5 introduces exceptions handling.

Interfaces

PHP5 introduces Interfaces.

E-Strict Error Level

PHP5 introduces new error level defined as E\_STRICT.

E\_STRICT will notify you when you use depreciated code.

Extensions

PHP5 introduces new default extensions.

Simple XML

DOM and XSL

PDO

Hash

- See more at: http://nextresources.blogspot.in/2012/11/php4-vs-php5-difference-between-php4.html#sthash.bZrtdX1r.dpuf

**MYSQL INDEXES:**

MySQL allows four general types of indexes (keys). These indexes can be created either on single column or multi-columns however both single column index and multi columns index have some different behaviors.

Primary Key Indexes  
Unique Key Indexes  
Normal Indexes also known as “Non-Unique Indexes”, “Ordinary Indexes”, or “Indexes without constraints”  
Full-Text Indexes

There are four types of statements for adding indexes to a table:

* **ALTER TABLE tbl\_name ADD PRIMARY KEY (column\_list):** This statement adds a PRIMARY KEY, which means that indexed values must be unique and cannot be NULL.
* **ALTER TABLE tbl\_name ADD UNIQUE index\_name (column\_list):** This statement creates an index for which values must be unique (with the exception of NULL values, which may appear multiple times).
* **ALTER TABLE tbl\_name ADD INDEX index\_name (column\_list):** This adds an ordinary index in which any value may appear more than once.
* **ALTER TABLE tbl\_name ADD FULLTEXT index\_name (column\_list):** This creates a special FULLTEXT index that is used for text-searching purposes.

Here is the example to add index in an existing table.