

Java Interview Questions

Meaning - Abstract classes, abstract methods

Difference - Java, C++

Difference between == and equals method

Explain Java security model

Explain working of Java Virtual Machine (JVM)

Difference : Java Beans, Servlets

Difference : AWT, Swing

Disadvantages of Java

What is BYTE Code ?

What gives java its "write once and run anywhere" nature?

Does Java have "goto"?

What is the meaning of "final" keyword?

Can I create final executable from Java?

Explain Garbage collection mechanism in Java

Why Java is not 100% pure object oriented language?

What are interfaces? or How to support multiple inheritance in Java?

How to use C++ code in Java Program?

Difference between "APPLET" and "APPLICATION"

Describe what happens when an object is created in Java

Several things happen in a particular order to ensure the object is constructed properly:

1. Memory is allocated from heap to hold all instance variables and implementation-specific data of the object and its superclasses. Implementation-specific data includes pointers to class and method data.
2. The instance variables of the objects are initialized to their default values.
3. The constructor for the most derived class is invoked. The first thing a constructor does is call the constructor for its superclasses. This process continues until the constructor for java.lang.Object is called, as java.lang.Object is the base class for all objects in java.
4. Before the body of the constructor is executed, all instance variable initializers and initialization blocks are executed. Then the body of the constructor is executed. Thus, the constructor for the base class completes first and constructor for the most derived class completes last.

In Java, You can create a String object as below :

```
String str = "abc"; & String str = new String("abc");
```

Why can't a button object be created as : Button bt = "abc"

Why is it compulsory to create a button object as: Button bt = new Button("abc");

Why this is not compulsory in String's case.

```
Button bt1 = "abc";
```

is because "abc" is a literal string (something slightly different than a String object, by-the-way) and bt1 is a Button object. That's simple. The only object in Java that can be assigned a literal String is java.lang.String.

Important to note that you are NOT calling a java.lang.String constructor when you type String s = "abc";

For example

```
String x = "abc";
```

```
String y = "abc";
```

refer to the same object.

While

```
String x1 = new String("abc");
```

```
String x2 = new String("abc");
```

refer to two different objects.

What is the advantage of OOP?

You will get varying answers to this question depending on whom you ask. Major advantages of OOP, IMHO, are:

1. simplicity: software objects model real world objects, so the complexity is reduced and the program structure is very clear;
2. modularity: each object forms a separate entity whose internal workings are decoupled from other parts of the system;
3. modifiability: it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods;
4. extensibility: adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones;
5. maintainability: objects can be maintained separately, making locating and fixing problems easier;
6. re-usability: objects can be reused in different programs

What are the main differences between Java and C++?

Everything is an object in Java(Single root hierarchy as everything gets derived from java.lang.Object)

Java does not have all the complicated aspects of C++ (For ex: Pointers, templates, unions, operator overloading, structures etc..)

The Java language promoters initially said "No pointers!", but when many programmers questioned how you can work without pointers, the promoters began saying "Restricted pointers." You can make up your mind whether it's really a pointer or not. In any event, there's no pointer arithmetic.

There are no destructors in Java. (automatic garbage collection)

Java does not support conditional compile (#ifdef/#ifndef type).

Thread support is built into java but not in C++.

Java does not support default arguments.

There's no scope resolution operator :: in Java. Java uses the dot for everything, but can get away with it since you can define elements only within a class. Even the method definitions must always occur within a class, so there is no need for scope resolution there either.

There's no "goto " statement in Java.

Java doesn't provide multiple inheritance (MI), at least not in the same sense that C++ does.

Exception handling in Java is different because there are no destructors.

Java has method overloading, but no operator overloading. The String class does use the + and += operators to concatenate strings and String expressions use automatic type conversion, but that's a special built-in case.

Java is interpreted for the most part and hence platform independent

What are interfaces?

Interfaces provide more sophisticated ways to organize and control the objects in your system. The interface keyword takes the abstract concept one step further. You could think of it as a “pure” abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but The interface keyword takes the abstract concept one step further. You could think of it as a “pure” abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields, but

An interface says: “This is what all classes that implement this particular interface will look like.” Thus, any code that uses a particular interface knows what methods might be called for that interface, and that’s all. So the interface is used to establish a “protocol” between classes. (Some object-oriented programming languages have a keyword called protocol to do the same thing.)

Typical example from "Thinking in Java":

```
import java.util.*;
interface Instrument {
    int i = 5; // static & final
    // Cannot have method definitions:
    void play(); // Automatically public
    String what();
    void adjust();
}
class Wind implements Instrument {
    public void play() {
        System.out.println("Wind.play()");
    }
    public String what() { return "Wind"; }
    public void adjust() {}
}
```

How can you achieve Multiple Inheritance in Java?

Java's interface mechanism can be used to implement multiple inheritance, with one important difference from c++ way of doing MI:

the inherited interfaces must be abstract. This obviates the need to choose between different implementations, as with interfaces there are no implementations.

example:

```
interface CanFight {
    void fight();
}
interface CanSwim {
    void swim();
}
interface CanFly {
    void fly();
}
class ActionCharacter {
    public void fight() {}
}
class Hero extends ActionCharacter implements CanFight, CanSwim, CanFly {
    public void swim() {}
    public void fly() {}
}
```

You can even achieve a form of multiple inheritance where you can use the *functionality* of classes rather than just the interface:

```
interface A {
```

```

void methodA();
}
class AImpl implements A {
void methodA() { //do stuff }
}
interface B {
void methodB();
}
class BImpl implements B {
void methodB() { //do stuff }
}
class Multiple implements A, B {
private A a = new A();
private B b = new B();
void methodA() { a.methodA(); }
void methodB() { b.methodB(); }
}

```

This completely solves the traditional problems of multiple inheritance in C++ where name clashes occur between multiple base classes. The coder of the derived class will have to explicitly resolve any clashes.

Don' t you hate people ~~who~~ point out minor typos?

Everything in the previous example is correct, except you need to instantiate an AImpl and BImpl. So class Multiple would look like this:

```

class Multiple implements A, B {
private A a = new AImpl();
private B b = new BImpl();
void methodA() { a.methodA(); }
void methodB() { b.methodB(); }
}

```

What is the difference between StringBuffer and String class?

A string buffer implements a mutable sequence of characters. A string buffer is like a String, but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.

The String class represents character strings. All string literals in Java programs, such as "abc" are constant and implemented as instances of this class; their values cannot be changed after they are created.

Strings in Java are known to be immutable. What it means is that every time you need to make a change to a String variable, behind the scene, a "new" String is actually being created by the JVM.

For an example:

if you change your String variable 2 times, then you end up with 3 Strings: one current and 2 that are ready for garbage collection.

The garbage collection cycle is quite unpredictable and these additional unwanted Strings will take up memory until that cycle occurs.

For better performance, use StringBuffers for string-type data that will be reused or changed frequently. There is more overhead per class than using String, but you will end up with less overall classes and consequently consume less memory.

Describe, in general, how java' s garbage collector works?

The Java runtime environment deletes objects when it determines that they are no longer being used. This process is known as garbage collection.

The Java runtime environment supports a garbage collector that periodically frees the memory used by objects that are no longer needed. The Java garbage collector is a mark-sweep garbage collector that scans Java' s dynamic memory areas for objects, marking those that are referenced. After all possible paths to objects are investigated, those objects that are not marked (i.e. are not referenced) are known to be garbage and are collected. (A more complete description of our garbage collection algorithm might be "A compacting, mark-sweep collector with some conservative scanning".)

The garbage collector runs synchronously when the system runs out of memory, or in response to a request from a Java program. Your Java program can ask the garbage collector to run at any time by calling `System.gc()`. The garbage collector requires about 20 milliseconds to complete its task so, your program should only run the garbage collector when there will be no performance impact and the program anticipates an idle period long enough for the garbage collector to finish its job.

Note: Asking the garbage collection to run does not guarantee that your objects will be garbage collected.

The Java garbage collector runs asynchronously when the system is idle on systems that allow the Java runtime to note when a thread has begun and to interrupt another thread (such as Windows 95). As soon as another thread becomes active, the garbage collector is asked to get to a consistent state and then terminate.

What' s the difference between `==` and equals method?

`equals` checks for the content of the string objects while `==` checks for the fact that the two String objects point to same memory location ie they are same references.

What are abstract classes, abstract methods?

Simply speaking a class or a method qualified with "abstract" keyword is an abstract class or abstract method.

You create an abstract class when you want to manipulate a set of classes through a common interface. All derived-class methods that match the signature of the base-class declaration will be called using the dynamic binding mechanism.

If you have an abstract class, objects of that class almost always have no meaning. That is, abstract class is meant to express only the interface and sometimes some default method implementations, and not a particular implementation, so creating an abstract class object makes no sense and are not allowed (compile will give you an error message if you try to create one).

An abstract method is an incomplete method. It has only a declaration and no method body. Here is the syntax for an abstract method declaration:
`abstract void f();`

If a class contains one or more abstract methods, the class must be qualified an abstract. (Otherwise, the compiler gives you an error message.). It's possible to create a class as abstract without including any abstract methods. This is useful when you've got a class in which it doesn't make sense to have any abstract methods, and yet you want to prevent any instances of that class. Abstract classes and methods are created because they make the abstractness of a class explicit, and tell both the user and the compiler how it was intended to be used.

For example:

```
abstract class Instrument {
```

```

int i; // storage allocated for each
public abstract void play();
public String what() {
return "Instrument";
public abstract void adjust();
}
class Wind extends Instrument {
public void play() {
System.out.println("Wind.play()");
}
public String what() { return "Wind"; }
public void adjust() {}

```

Abstract classes are classes for which there can be no instances at run time. i.e. the implementation of the abstract classes are not complete.

Abstract methods are methods which have no definition. i.e. abstract methods have to be implemented in one of the sub classes or else that class will also become Abstract.

What is the difference between an Applet and an Application?

A Java application is made up of a main() method declared as public static void that accepts a string array argument, along with any other classes that main() calls. It lives in the environment that the host OS provides.

A Java applet is made up of at least one public class that has to be subclassed from java.awt.Applet. The applet is confined to living in the user' s Web browser, and the browser' s security rules, (or Sun' s appletviewer, which has fewer restrictions).

The differences between an applet and an application are as follows:

1. Applets can be embedded in HTML pages and downloaded over the Internet whereas Applications have no special support in HTML for embedding or downloading.
2. Applets can only be executed inside a java compatible container, such as a browser or appletviewer whereas Applications are executed at command line by java.exe or jview.exe.
3. Applets execute under strict security limitations that disallow certain operations(sandbox model security) whereas Applications have no inherent security restrictions.
4. Applets don' t have the main() method as in applications. Instead they operate on an entirely different mechanism where they are initialized by init(),started by start(),stopped by stop() or destroyed by destroy().

Java says "write once, run anywhere". What are some ways this isn' t quite true?

As long as all implementaions of java are certified by sun as 100% pure java this promise of "Write once, Run everywhere" will hold true. But as soon as various java core implemenations start digressing from each other, this won' t be true anymore. A recent example of a questionable business tactic is the surreptitious behavior and interface modification of some of Java' s core classes in their own implementation of Java. Programmers who do not recognize these undocumented changes can build their applications expecting them to run anywhere that Java can be found, only to discover that their code works only on Microsoft' s own Virtual Machine, which is only available on Microsoft' s own operating systems.

What is the difference between a Vector and an Array. Discuss the advantages and disadvantages of both?

Vector can contain objects of different types whereas array can contain objects only of a single type.

- Vector can expand at run-time, while array length is fixed.
- Vector methods are synchronized while Array methods are not

What are java beans?

JavaBeans is a portable, platform-independent component model written in the Java programming language, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere -- benefiting from the platform-independent power of Java technology. JavaBeans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container applications.

JavaBeans are usual Java classes which adhere to certain coding conventions:

1. Implements java.io.Serializable interface
2. Provides no argument constructor
3. Provides getter and setter methods for accessing it's properties

What is RMI?

RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do.

Above excerpt is from "Thinking in java". For more information refer to any book on Java.

What does the keyword "synchronize" mean in java. When do you use it? What are the disadvantages of synchronization?

Synchronize is used when u want to make ur methods thread safe. The disadvantage of synchronise is it will end up in slowing down the program. Also if not handled properly it will end up in dead lock.

Any more additions to this

What gives java it's "write once and run anywhere" nature?

Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platform specific and hence can be fed to any platform. After being fed to the JVM, which is specific to a particular operating system, the code platform specific machine code is generated thus making java platform independent.

What are native methods? How do you use them?

Native methods are methods written in other languages like C, C++, or even assembly language. You can call native methods from Java using JNI.

Native methods are used when the implementation of a particular method is present in language other than Java say C, C++.

To use the native methods in java we use the keyword native

```
public native method_a()
```

This native keyword is signal to the java compiler that the implementation of this method is in a language other than java.

Native methods are used when we realize that it would take up a lot of rework to write that piece of already existing code in other language to java.

What is JDBC? Describe the steps needed to execute a SQL query using JDBC.

We can connect to databases from java using JDBC. It stands for Java DataBase Connectivity.

Here are the steps:

1. Register the jdbc driver with the driver manager
2. Establish jdbc connection
3. Execute an sql statement
4. Process the results
5. Close the connection

Before doing these do import java.sql.*

JDBC is java based API for accessing data from the relational databases. JDBC provides a set of classes and interfaces for doing various database operations.

The steps are:

Register/load the jdbc driver with the driver manager.

Establish the connection thru DriverManager.getConnection();

Fire a SQL thru conn.executeStatement();

Fetch the results in a result set

Process the results

Close statement/result set and connection object.

How many different types of JDBC drivers are present? Discuss them.

There are four JDBC driver types.

Type 1: JDBC-ODBC Bridge plus ODBC Driver:

The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

Type 2: Native-API partly-Java Driver:

The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.

Type 3: JDBC-Net Pure Java Driver:

The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

Type 4: Native-Protocol Pur Java Driver

These are pure Java drivers that communicate directly with the vendor' s database. They do this by converting JDBC commands directly into the database engine' s native protocol. This driver has no additional translation or middleware layer, which improves performance tremendously.

What does the "static" keyword mean in front of a variable? A method? A class? Curly braces {}?

static variable

- means a class level variable

static method:

-does not have "this". It is not allowed to access the not static members of the class.

can be invoked enev before a single instance of a class is created.

eg: main

static class:

no such thing.

static free floating block:

is executed at the time the class is loaded. There can be multiple such blocks. This may be useful to load native libraries when using native methods.

eg:

```
native void doThis(){
```

```
static{
```

```
System.loadLibrary("myLibrary.lib");
```

```
.....
```

```
}
```

Access specifiers: "public", "protected", "private", nothing?

Unfortunately I believe that buturab is discussing Static vs. non-Static methods and variables. In the case of Public, Private and Protected, that is used to describe which programs can access that class or method:

Public – any other class from any package can instantiate and execute the classes and methods

Protected – only subclasses and classes inside of the package can access the classes and methods

Private – the original class is the only class allowed to executed the methods.

What does the "final" keyword mean in front of a variable? A method? A class?

FINAL for a variable : value is constant

FINAL for a method : cannot be overridden

FINAL for a class : cannot be derived

A final variable cannot be reassigned,

but it is not constant. For instance,

```
final StringBuffer x = new StringBuffer()
```

```
x.append("hello");
```

is valid. X cannot have a new value in it, but nothing stops operations on the object that it refers, including destructive operations. Also, a final method cannot be overridden or hidden by new access specifications. This means that the compiler can choose to in-line the invocation of such a method. (I don't know if any compiler actually does this, but it's true in theory.) The best example of a final class is String, which defines a class that cannot be derived.

Does Java have "goto"?

No.

Why "bytecode"? Can you reverse-engineer the code from bytecode?

you can get the sourcecode from ur class file.

u get a free JAD here

What synchronization constructs does Java provide? How do they work?

The two common features that are used are:

1. Synchronized keyword - Used to synchronize a method or a block of code. When you synchronize a method, you are in effect synchronizing the code within the method using the monitor of the current object for the lock.

The following have the same effect.

```
synchronized void foo() {
```

```
}
```

```

and
void foo() {
synchronized(this) {
}
}

```

If you synchronize a static method, then you are synchronizing across all objects of the same class, i.e. the monitor you are using for the lock is one per class, not one per object.

2. wait/notify. wait() needs to be called from within a synchronized block. It will first release the lock acquired from the synchronization and then wait for a signal. In Posix C, this part is equivalent to the pthread_cond_wait method, which waits for an OS signal to continue. When somebody calls notify() on the object, this will signal the code which has been waiting, and the code will continue from that point. If there are several sections of code that are in the wait state, you can call notifyAll() which will notify all threads that are waiting on the monitor for the current object. Remember that both wait() and notify() have to be called from blocks of code that are synchronized on the monitor for the current object.

Does Java have multiple inheritance?

Java does not support multiple inheritance directly but it does thru the concept of interfaces.

We can make a class implement a number of interfaces if we want to achieve multiple inheritance type of functionality of C++.

How does exception handling work in Java?

1. It separates the working/functional code from the error-handling code by way of try-catch clauses.

2. It allows a clean path for error propagation. If the called method encounters a situation it can't manage, it can throw an exception and let the calling method deal with it.

3. By enlisting the compiler to ensure that "exceptional" situations are anticipated and accounted for, it enforces powerful coding.

4. Exceptions are of two types:

Compiler-enforced exceptions, or checked exceptions

Runtime exceptions, or unchecked exceptions

Compiler-enforced (checked) exceptions are instances of the Exception class or one of its subclasses -- excluding the RuntimeException branch. The compiler expects all checked exceptions to be appropriately handled. Checked exceptions must be declared in the throws clause of the method throwing them -- assuming, of course, they're not being caught within the same method. The calling method must take care of these exceptions by either catching or declaring them in its throws clause. Thus, making an exception checked forces us to pay heed to the possibility of it being thrown. An example of a checked exception is java.io.IOException. As the name suggests, it throws whenever an input/output operation is abnormally terminated.

Does Java have destructors?

garbage collector does the job working in the background

Java does not have destructors; but it has finalizers that do a similar job.

the syntax is

```

public void finalize(){
}

```

if an object has a finalizer, the method is invoked before the system garbage collects the object

What does the "abstract" keyword mean in front of a method? A class?

Abstract keyword declares either a method or a class.

If a method has a `abstract` keyword in front of it, it is called abstract method. Abstract method has no body. It has only arguments and return type. Abstract methods act as placeholder methods that are implemented in the subclasses.

Abstract classes can't be instantiated. If a class is declared as abstract, no objects of that class can be created. If a class contains any abstract method it must be declared as abstract.

Are Java constructors inherited? If not, why not?

You cannot inherit a constructor. That is, you cannot create an instance of a subclass using a constructor of one of its superclasses. One of the main reasons is because you probably don't want to override the superclass's constructor, which would be possible if they were inherited. By giving the developer the ability to override a superclass's constructor you would erode the encapsulation abilities of the language.

SQL:

What are two methods of retrieving SQL?

What cursor type do you use to retrieve multiple recordsets?

What is the difference between a "where" clause and a "having" clause?

"Where" is a kind of restriction statement. You use where clause to restrict all the data from DB. Where clause is used before result retrieving. But Having clause is used after retrieving the data. Having clause is a kind of filtering command.

What is the basic form of a SQL statement to read data out of a table?

The basic form to read data out of table is

```
SELECT * FROM table_name;
```

An answer:

```
SELECT *
```

```
FROM table_name
```

```
WHERE xyz= ' whatever' ;
```

cannot be called basic form because of WHERE clause.

What structure can you have the database make to speed up table reads?

The question is not correct.

"What structure can you have the database make to speed up table reads?"

It is not clear what exactly the term

"structure" means in this case.

Follow the rules of DB tuning we have to:

1] properly use indexes (different types of indexes)

2] properly locate different DB objects

across different tablespaces, files and so on.

3] create a special space (tablespace) to locate some of the data with special datatype

(for example CLOB, LOB and ...)

4

What are the tradeoffs with having indexes?

1. Faster selects, slower updates.

2. Extra storage space to store indexes.

updates are slower because in addition to updating the table you have to update the index.

What is a "join"?

' join' used to connect two or more tables logically with or without common field.

What is "normalization"? "Denormalization"? Why do you sometimes want to denormalize?

Normalizing data means eliminating redundant information from a table and organizing the data so that future changes to the table are easier.

Denormalization means allowing redundancy in a table.

The main benefit of denormalization is improved performance with simplified data retrieval and manipulation. This is done by reduction in the number of joins needed for data processing.

What is a "constraint"?

A constraint allows you to apply simple referential integrity checks to a table.

There are four primary types of constraints that are currently supported by SQL Server:

PRIMARY/UNIQUE - enforces uniqueness of a particular table column.

DEFAULT - specifies a default value for a column in case an insert operation does not provide one.

FOREIGN KEY - validates that every value in a column exists in a column of another table.

CHECK - checks that every value stored in a column is in some specified list

Each type of constraint performs a specific type of action.

Default is not a constraint.

NOT NULL is one more constraint which does not allow values in the specific column to be null.

And also it is the only constraint which is not a table level constraint.

What types of index data structures can you have?

An index helps to faster search values in tables. The three most commonly used index-types are:

- B-Tree: builds a tree of possible values with a list of row IDs that have the leaf value. Needs a lot of space and is the default index type for most databases.

- Bitmap: string of bits for each possible value of the column. Each bit string has one bit for each row. Needs only few space and is very fast.(however, domain of value cannot be large, e.g.

SEX(m,f); degree(BS,MS,PHD)

- Hash: A hashing algorithm is used to assign a set of characters to represent a text string such as a composite of keys or partial keys, and compresses the underlying data. Takes longer to build and is supported by relatively few databases.

What is a "primary key"?

A PRIMARY INDEX or PRIMARY KEY is something which comes mainly from database theory. From its behaviour is almost the same as an UNIQUE INDEX, i.e. there may only be one of each value in this column. If you call such an INDEX PRIMARY instead of UNIQUE, you say something about your table design, which I am not able to explain in few words.

Primary Key is a type of a constraint enforcing uniqueness and data integrity for each row of a table. All columns participating in a primary key constraint must possess the NOT NULL property.

What is a "functional dependency"? How does it relate to database table design?

Functional dependency relates to how one object depends upon the other in the database.

for example, procedure/function sp2 may be called by procedure sp1. Then we say that sp1 has

functional dependency on sp2.

What is a "trigger"?

Triggers are stored procedures created in order to enforce integrity rules in a database. A trigger is executed every time a data-modification operation occurs (i.e., insert, update or delete).

Triggers are executed automatically on occurrence of one of the data-modification operations.

A trigger is a database object directly associated with a particular table. It fires whenever a specific statement/type of statement is issued against that table. The types of statements are insert, update, delete and query statements. Basically, trigger is a set of SQL statements

2 of 3 people found this post useful.

A trigger is a solution to the restrictions of a constraint.

For instance :

1. A database column cannot carry PSEUDO columns as criteria where a trigger can.

2. A database constraint cannot refer old and new values for a row where a trigger can.

Why can a "group by" or "order by" clause be expensive to process?

Processing of "group by" or "order by" clause often requires creation of Temporary tables to process the results of the query.

Which depending of the result set can be very expensive.

What is "index covering" of a query?

Index covering means that "Data can be found only using indexes, without touching the tables"

Bottom of Form 1

Top of Form 3

Bottom of Form 3

What types of join algorithms can you have?

Nested loop, indexes and hash.

Top of Form 1

4 of 5 people found this post useful.

There are four type of joins

(join conditions):

1) equa join

2) non-equa join

3) self-join

4) outer join.

State some advantages AND disadvantages of indexing database tables.

This is the answer to advantage and disadvantage of Indexes:

Adv:

Faster querying of data

Disad:

1. Slower Insert

2. Slower Updates if you are also updating the primary key.

What is a SQL view?

An output of a query can be stored as a view. View acts like small table which meets our criterion.

Top of Form 1

6 of 6 people found this post useful.

View is a precompiled SQL query which is used to select data from one or more tables. A view is like a table but it doesn't physically take any space. View is a good way to present data in a

particular format if you use that query quite often.

View can also be used to restrict users from accessing the tables directly.