

java.interview.preparation

This blog is to provide details for java interview preparation. Everybody is busy in day to day project work and most of times we are not able to spend time due to workload. I am trying to provide the consolidated details of java topics and sharing my java interview experiences.

Home Concurrency Design Pattern Spring MVC Spring JMS Data structure Collection Threading Hibernate Face 2 Fa
Interview Planning Interview Questions Memory Mgmt & GC XML Puzzles DB Java 1.5 Transaction Inner Class Cache

Database Interview Questions for Java Developers



'Java database questions' or 'Database interview questions for java developer' is covered in this blog. All enterprise applications uses RDBS to persist data and latter use them for business processing . All java developer must looked into database to figure out the issues and persistence of business data. The expectation from java developer is - they should able to analysis the [application data](#) issues and perform [basic](#) database operation to support and develop the better application.

Database Questions - Part 2

Describe Oracle database's physical and logical structure ?

Physical : Data files, Redo Log files, Control file.

Logical : Tables, [Views](#), Tablespaces, etc.

Can a view be updated/inserted/deleted? If Yes - under what conditions ?

A View can be updated/deleted/inserted if it has only one [base table](#) if the view is based on columns from one or more tables then insert, update and delete is not possible.

What is explain plan and how is it used ?

The EXPLAIN PLAN command is a tool to tune SQL statements. To use it you must have an explain_table generated in the user you are running the explain plan for. This is created using the utlxplan.sql script. Once the explain plan table exists you run the explain plan command giving as its argument the SQL statement to be explained. The explain_plan table is then queried to see the execution plan of the statement. Explain plans can also be run using tkprof.

What is a mutating table error and how can you get around it ?

This happens with triggers. It occurs because the trigger is trying to update a row it is currently using. The usual fix involves either use of views or temporary tables so the database is selecting from one while updating the other.

Describe the use of %ROWTYPE and %TYPE in PL/SQL ?

%ROWTYPE allows you to associate a variable with an entire table row. The %TYPE associates a variable with a single column type.

Explain the difference between a data block, an extent and a segment ?

A **data block** is the smallest unit of logical storage for a database object. As objects grow they take chunks of additional storage that are composed of [contiguous](#) data blocks. These groupings of contiguous data blocks are called **extents**. All the extents that an object takes when grouped together are considered the **segment** of the database object.

Explain the difference between a FUNCTION, PROCEDURE and PACKAGE ?

A function and procedure are the same in that they are intended to be a collection of PL/SQL code that carries a single task. While a procedure does not have to return any values to the calling application, a function will return a single value. A package on the other hand is a collection of functions and procedures that are grouped together based on their commonality to a [business function](#) or application.

Explain materialized views and how they are used ?

Materialized views are objects that are reduced sets of information that have been summarized, grouped, or aggregated from [base tables](#). They are typically used in data warehouse or decision support systems. The **JobQueue process** refresh the materialized view data.

Explain Database Pessimistic and Optimistic locking?

Transactional isolation is usually implemented by locking whatever is accessed in a transaction. There are two different approaches to transactional locking:

1. Pessimistic locking
2. Optimistic locking

The disadvantage of pessimistic locking is that a [resource](#) is locked from the time it is first accessed in a transaction until the transaction is finished,

making it inaccessible to other transactions during that time. If most transactions simply look at the resource and never change it, an exclusive lock may be overkill as it may cause lock contention, and optimistic locking may be a better approach. With pessimistic locking, locks are applied in a fail-safe way. In the [banking application](#) example, an account is locked as soon as it is accessed in a transaction. Attempts to use the account in other transactions while it is locked will either result in the other process being delayed until the account lock is released, or that the process transaction will be rolled back. The lock exists until the transaction has either been committed or rolled back.

With optimistic locking, a resource is not actually locked when it is first accessed by a transaction. Instead, the state of the resource at the time when it would have been locked with the pessimistic locking approach is saved. Other transactions are able to concurrently access to the resource and the possibility of conflicting changes is possible. At commit time, when the resource is about to be updated in persistent storage, the state of the resource is read from storage again and compared to the state that was saved when the resource was first accessed in the transaction. If the two states differ, a conflicting update was made, and the transaction will be rolled back. This is very important for [application performance](#) to use Optimistic locking.

Difference in Inner join and Outer join?

Inner Joins

An **inner join** (sometimes called a **simple join**) is a join of two or more tables that returns only those rows that satisfy the join condition.

Outer Joins

An **outer join** extends the result of a simple join. An outer join returns all rows that satisfy the join condition and also returns some or all of those rows from one table for which no rows from the other satisfy the join condition.

- To write a query that performs an outer join of tables A and B and returns all rows from A (a **left outer join**), use the LEFT [OUTER] JOIN syntax in the FROM clause, or apply the outer join operator (+) to all columns of B in the join condition in the WHERE clause. For all rows in A that have no matching rows in B, Oracle Database returns null for any select list expressions containing columns of B.
- To write a query that performs an outer join of tables A and B and returns all rows from B (a **right outer join**), use the RIGHT [OUTER] JOIN syntax in the FROM clause, or apply the outer join operator (+) to all columns of A in the join condition in the WHERE clause. For all rows in B that have no matching rows in A, Oracle returns null for any select list expressions containing columns of A.

To write a query that performs an outer join and returns all rows from A and B, extended with nulls if they do not satisfy the join condition (a **full outer join**), use the FULL [OUTER] JOIN syntax in the FROM clause.

Why Group by is only used with having clause and aggregative functions example?

HAVING is used to perform an action on groups created by GROUP BY similar to that of the WHERE clause on rows in a basic SQL statement. The WHERE clause limits the rows evaluated. The HAVING clause limits the grouped rows returned.

- **AVG**: returns the variance or variability of an expression.
- **COUNT**: returns the number of rows returned by a query.
- **FIRST**: returns the first value from an ordered sequence.
- **LAST**: returns the last value from an ordered sequence.
- **MAX**: returns the maximum value of a column.
- **MIN**: returns the minimum value of a column.
- **STDDEV**: returns the standard deviation of a set of numbers.
- **SUM**: returns the sum or total the values of a column.
- **VARIANCE**: returns the variance or *variability* of an expression

HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause.

A HAVING clause is like a WHERE clause, but applies only to groups as a whole, whereas the WHERE clause applies to [individual](#) rows. A query can contain both a WHERE clause and a HAVING clause. The WHERE clause is applied first to the individual rows in the tables. Only the rows that meet the conditions in the WHERE clause are grouped. The HAVING clause is then applied to the rows in the result set. Only the groups that meet the HAVING conditions appear in the query output. You can apply a HAVING clause only to columns that also appear in the GROUP BY clause or in an aggregate function. **Example of HAVING and WHERE in one query:**

```
SELECT titles.pub_id, AVG( titles.price)
FROM titles INNER JOIN publishers ON titles.pub_id = publishers.pub_id WHERE publishers.state = 'CA' GROUP BY titles.pub_id HAVING
AVG(titles.price) > 10;
```

Sometimes you can specify the same set of rows using either a WHERE clause or a HAVING clause. In such cases, one method is not more or less efficient than the other. The optimizer always automatically analyzes each statement you enter and selects an efficient means of executing it. It is best to use the syntax that most clearly describes the desired result. In general, that means eliminating undesired rows in earlier clauses.

What is a Stored Procedure?

A stored procedure is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input data. And when the procedure is modified, all clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database.

e.g. sp_helpdb, sp_renamedb, sp_depends etc.

What is a Trigger?

A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs. Triggers are stored in and managed by the DBMS. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed; DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers can be considered to be similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-driven and are not attached to a specific table as triggers are. Stored procedures are explicitly executed by invoking a CALL to the procedure while triggers are implicitly executed. In addition, triggers can also execute stored procedures.

Nested Trigger: A trigger can also contain INSERT, UPDATE and DELETE logic within itself; so when the trigger is fired because of data modification, it can also cause another data modification, thereby firing another trigger. A trigger that contains data modification logic within itself is called a nested trigger.

What are the Different Types of Triggers?

There are two types of Triggers.

1) DML Trigger

There are two types of DML Triggers

1. Instead of Trigger

Instead of Triggers are fired in place of the triggering action such as an insert, update, or delete.

2. After Trigger

After triggers execute following the triggering action, such as an insert, update, or delete.

2) DDL Trigger

This type of trigger is fired against Drop Table, Create Table, Alter Table or Login events. DDL Triggers are always After Triggers.

What is a View?

A simple view can be thought of as a subset of a table. It can be used for retrieving data as well as updating or deleting rows. Rows updated or deleted in the view are updated or deleted in the table the view was created with. It should also be noted that as data in the original table changes, so does the data in the view as views are the way to look at parts of the original table. The results of using a view are not permanently stored in the database. The data accessed through a view is actually constructed using standard T-SQL select command and can come from one to many different base tables or even other views.

What is an Index?

An index is a physical structure containing pointers to the data. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries. Effective indexes are one of the best ways to improve performance in a database application. A table scan happens when there is no index available to help a query. In a table scan, the SQL Server examines every row in the table to satisfy the query results. Table scans are sometimes unavoidable, but on large tables, scans have a terrific impact on performance.

What is a Cursor?

A cursor is a database object used by applications to manipulate data in a set on a row-by-row basis, instead of the typical SQL commands that operate on all the rows in the set at one time.

In order to work with a cursor, we need to perform some steps in the following order:

- Declare cursor
- Open cursor
- Fetch row from the cursor
- Process fetched row
- Close cursor
- Deallocate cursor.

What is the Difference between a Function and a Stored Procedure?

UDF can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section, whereas Stored procedures cannot be. UDFs that return tables can be treated as another rowset. This can be used in JOINS with other tables. Inline UDF's can be thought of as views that take parameters and can be used in JOINS and other Rowset operations.

What are Different Types of Join?

Cross Join

A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. The common example is when company wants to combine each product with a pricing table to analyze each product at each price.

Inner Join

A join that displays only the rows that have a match in both joined tables is known as inner Join. This is the default type of join in the Query and View Designer.

Outer Join

- A join that includes rows even if they do not have related rows in the joined table is an Outer Join. You can create three different outer join to specify the unmatched rows to be included:
- Left Outer Join: In Left Outer Join, all the rows in the first-named table, i.e. "left" table, which appears leftmost in the JOIN clause, are included. Unmatched rows in the right table do not appear.
- Right Outer Join: In Right Outer Join, all the rows in the second-named table, i.e. "right" table, which appears rightmost in the JOIN clause are included. Unmatched rows in the left table are not included.
- Full Outer Join: In Full Outer Join, all the rows in all joined tables are included, whether they are matched or not.

Self Join

This is a particular case when one table joins to itself with one or two aliases to avoid confusion. A self join can be of any type, as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. The common example is when company has a hierarchal reporting structure whereby one member of staff reports to another. Self Join can be Outer Join or Inner Join.

What is the Correct Order of the Logical Query Processing Phases?

The correct order of the Logical Query Processing Phases is as follows:

1. FROM
2. ON
3. OUTER
4. WHERE
5. GROUP BY
6. CUBE | ROLLUP
7. HAVING
8. SELECT
9. DISTINCT
10. TOP

11. ORDER BY

Difference in clustered and unclustered index?**Non-clustered**

The data is present in arbitrary order, but the logical ordering is specified by the index. The data rows may be spread throughout the table regardless of the value of the indexed column or expression. The non-clustered index tree contains the index keys in sorted order, with the leaf level of the index containing the pointer to the record (page and the row number in the data page in page-organized engines; row offset in file-organized engines). In a non-clustered index: The physical order of the rows is not the same as the index order. Typically created on non-primary key columns used in JOIN, WHERE, and ORDER BY clauses. There can be more than one non-clustered index on a database table.

Clustered

Clustering alters the data block into a certain distinct order to match the index, resulting in the row data being stored in order. Therefore, only one clustered index can be created on a given database table. Clustered indices can greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected. Since the physical records are in this sort order on disk, the next row item in the sequence is immediately before or after the last one, and so fewer data block reads are required. The primary feature of a clustered index is therefore the ordering of the physical data rows in accordance with the index blocks that point to them. Some databases separate the data and index blocks into separate files, others put two completely different data blocks within the same physical file(s).

What are the Different Index Configurations a Table can have?

A table can have one of the following index configurations:

- No indexes
- A clustered index
- A clustered index and many non-clustered indexes
- A non-clustered index
- Many non-clustered indexes

What's the Difference between a Primary Key and a Unique Key?

Both primary key and unique key enforce uniqueness of the column on which they are defined. But by default, the primary key creates a clustered index on the column, whereas unique key creates a non-clustered index by default. Another major difference is that primary key doesn't allow NULLs, but unique key allows one NULL only.

What is Difference between DELETE and TRUNCATE Commands?

Delete command removes the rows from a table on the basis of the condition that we provide with a WHERE clause. Truncate will actually remove all the rows from a table, and there will be no data in the table after we run the truncate command.

TRUNCATE

- TRUNCATE is faster and uses fewer system and transaction log resources than DELETE. (Read all the points below)
- TRUNCATE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log.
- TRUNCATE removes all the rows from a table, but the table structure, its columns, constraints, indexes and so on remains. The counter used by an identity for new rows is reset to the seed for the column.
- You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint.
- Using T-SQL – TRUNCATE cannot be rolled back unless it is used in TRANSACTION. OR TRUNCATE can be rolled back when used with BEGIN ... END TRANSACTION using T-SQL.
- TRUNCATE is a DDL Command.
- TRUNCATE resets the identity of the table.

DELETE

- DELETE removes rows one at a time and records an entry in the transaction log for each deleted row.
- DELETE does not reset Identity property of the table.
- DELETE can be used with or without a WHERE clause
- DELETE activates Triggers if defined on table.
- DELETE can be rolled back.
- DELETE is DML Command.
- DELETE does not reset the identity of the table.

What are Different Types of Locks?

- Shared Locks: Used for operations that do not change or update data (read-only operations), such as a SELECT statement.
- Update Locks: Used on resources that can be updated. It prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and potentially updating resources later.
- Exclusive Locks: Used for data-modification operations, such as INSERT, UPDATE, or DELETE. It ensures that multiple updates cannot be made to the same resource at the same time.

What are Pessimistic Lock and Optimistic Lock?

Optimistic Locking is a strategy where you read a record, take note of a version number and check that the version hasn't changed before you write the record back. If the record is dirty (i.e. different version to yours), then you abort the transaction and the user can re-start it.

Pessimistic Locking is when you lock the record for your exclusive use until you have finished with it. It has much better integrity than optimistic locking but requires you to be careful with your application design to avoid Deadlocks.

When is the use of UPDATE_STATISTICS command?

This command is basically used when a large amount of data is processed. If a large amount of deletions, modifications or Bulk Copy into the tables has occurred, it has to update the indexes to take these changes into account. UPDATE_STATISTICS updates the indexes on these tables accordingly.

What is the Difference between a HAVING clause and a WHERE clause?

They specify a search condition for a group or an aggregate. But the difference is that HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query, whereas WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.

What is CHECK Constraint?

A CHECK constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity.

What is NOT NULL Constraint?

A NOT NULL constraint enforces that the column will not accept null values. The not null constraints are used to enforce domain integrity, as the check constraints.

What is the difference between UNION and UNION ALL?**UNION**

The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type. With UNION, only distinct values are selected.

UNION ALL

The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values. The difference between UNION and UNION ALL is that UNION ALL will not eliminate duplicate rows, instead it just pulls all rows from all the tables fitting your query specifics and combines them into a table.

What is B-Tree?

The database server uses a B-tree structure to organize index information. B-Tree generally has following types of index pages or nodes:

- *Root node: A root node contains node pointers to only one branch node.*
- *Branch nodes: A branch node contains pointers to leaf nodes or other branch nodes, which can be two or more.*
- *Leaf nodes: A leaf node contains index items and horizontal pointers to other leaf nodes, which can be many.*

What are the Advantages of Using Stored Procedures?

- *Stored procedure can reduced network traffic and latency, boosting application performance.*
- *Stored procedure execution plans can be reused; they staying cached in SQL Server's memory, reducing server overhead.*
- *Stored procedures help promote code reuse.*
- *Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.*
- *Stored procedures provide better security to your data.*

What is BCP? When is it Used?

BCP or BulkCopy is a tool used to copy huge amounts of data from tables and views. BCP does not copy the complete structures from source to destination. BULK INSERT command helps to import a data file into a database table or view in a user-specified format.

What is SQL Injection? How to Protect Against SQL Injection Attack?

SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

Here are few methods which can be used to protect again SQL Injection attack:

- Use Type-Safe SQL Parameters
- Use Parameterized Input with Stored Procedures
- Use the Parameters Collection with Dynamic SQL
- Filtering Input parameters
- Use the escape character in LIKE clause
- Wrapping Parameters with QUOTENAME() and REPLACE()

What is CHECKPOINT Process in the SQL Server?

CHECKPOINT process writes all dirty pages for the current database to disk. Dirty pages are data pages that have been entered into the buffer cache and modified, but not yet written to disk.

What special Oracle feature allows you to specify how the cost based system treats a SQL statement?

The COST based system allows the use of HINTs to control the optimizer path selection. If they can give some example hints such as FIRST ROWS, ALL ROWS, USING INDEX, STAR, even better.

What is a Hint?

Hints are options and strong suggestions specified for enforcement by the SQL Server query processor on DML statements. The hints override any execution plan the query optimizer might select for a query

There are three different types of hints. Let us understand the basics of each of them separately.

Join Hint

This hint is used when more than one table is used in a query. Two or more tables can be joined using different types of joins. This hint forces the type of join algorithm that is used. Joins can be used in SELECT, UPDATE and DELETE statements.

Query Hint

This hint is used when certain kind of logic has to be applied to a whole query. Any hint used in the query is applied to the complete query as opposed

to a part of it. There is no way to specify that only a certain part of a query should be used with the hint. After any query, the `OPTION` clause is specified to apply the logic to this query. A query always has any of the following statements: `SELECT`, `UPDATE`, `DELETE`, `INSERT` or `MERGE` (SQL 2K8); and this hint can be applied to all of them.

Example:

```
SELECT /*+ INDEX(i dcf_vol_prospect_ids_idx)*/
      /*+ LEADING(i vol) */
      /*+ ALL_ROWS */
      i.id_number,
      ...
FROM i_table i
JOIN vol_table vol on vol.id_number = i.id_number
JOIN to_a_bunch_of_other_tables...
WHERE i.solicitor_id = '123'
      AND vol.solicitable_ind = 1;
```

What is Database Partitioning?

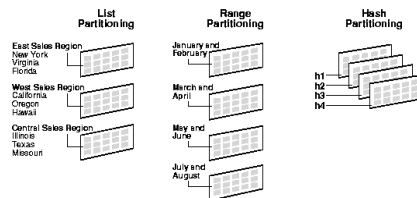
Partitioning addresses key issues in supporting very large tables and indexes by letting you decompose them into smaller and more manageable pieces called **partitions**. SQL queries and DML statements do not need to be modified in order to access partitioned tables. However, after partitions are defined, DDL statements can access and manipulate individual partitions rather than entire tables or indexes. This is how partitioning can simplify the manageability of large database objects. Also, partitioning is entirely transparent to applications.

Each partition of a table or index must have the same logical attributes, such as column names, datatypes, and constraints, but each partition can have separate physical attributes such as `pctfree`, `pctused`, and `tablespaces`.

Partitioning is useful for many different types of applications, particularly applications that manage large volumes of data. OLTP systems often benefit from improvements in manageability and availability, while data warehousing systems benefit from performance and manageability.

Oracle provides the following partitioning methods:

- Range Partitioning
- List Partitioning
- Hash Partitioning
- Composite Partitioning



Example:

```
create table Book (
  ID          number not null,
  TITLE       varchar2,
  PUBID       char,
  PUBDATE     date,
  constraint lab6_pk primary key(ID))
Partition by range (pubdate)(
  Partition p1 values less than (to_date('01-JAN-2000','DD-MON-YYYY')),
  Partition p2 values less than (to_date('01-JAN-2010','DD-MON-YYYY')),
  Partition p3 values less than (MAXVALUE)
)
```

You would insert and select data just the same way that you would for a non-partitioned table

```
INSERT INTO Book( id, title, pubid, pubdate )VALUES( 1, 'Something', 'FOO', sysdate );
```

```
SELECT rowid, id, title, pubid, pubdat FROM Book;
```

More Database Questions - Part 2

Java Jobs

[Hadoop Spark Developer Mechelen...](#)

Mechelen, Antwerpen, Belgium
Amoria Bond Ltd

[Software Entwickler/Developer \(m/w\)...](#)

München, Bayern, Germany
Darwin Recruitment

[SAP BI / BW Berater \(m/w\)](#)

München, Bayern, Germany
duerenhoff GmbH

[IT PROJECT MANAGER](#)

Baltimore, MD

DB Jobs

[Software Developer in Test Intern](#)

Toronto, ON, Canada
Street Context

[Big Data Engineer...](#)

Amsterdam, Noord-Holland, Netherlands
Astbury Marsden

[Senior Systems Database Administrator...](#)

Den Haag, Zuid-Holland, Netherlands
Next Ventures Ltd

[Lead SQL Server DBA](#)

Basilidon, Essex, United Kingdom

Testing job

[Database Programmer@Phoenix,...](#)

Phoenix, AZ
Sage IT INC

[Java Developer C++ Core Java J2SE...](#)

Newark, NJ
Joseph Harry Ltd

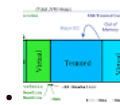
[Entwicklungsingenieur \(m/w\) Software](#)

Heidenheim, Bayern, Germany
FERCHAU Engineering GmbH

[ACF2 Specialist](#)

Gloucester, Gloucestershire, United...

<p>Synectics</p> <p>Java Software Engineer Arnhem ...</p> <p>Arnhem, Gelderland, Netherlands</p> <p>Ardekay</p> <p>POST A JOB ></p>	<p>Xchanging UK Ltd</p> <p>Microsoft Business Intelligence...</p> <p>Almere, Flevoland, Netherlands</p> <p>Claremont Consulting Ltd</p> <p>POST A JOB ></p>	<p>Pontoon</p> <p>PL/SQL Developer/Oracle DBA</p> <p>La Hulpe, Brabant Wallon, Belgium</p> <p>Volt</p> <p>POST A JOB ></p>
POWERED BY JOBTREAD	POWERED BY JOBTREAD	POWERED BY JOBTREAD



Java Memory Management
We are covering here -'Java garbage collection interview questions' or 'Java memory interview questions' in d...

• **Collection Interview Questions for Java Developers**
Java util.collections is one of the most important package in java and the important of this package can be understand by...

• **Java Concurrency**
Java Concurrency interview question - In year 2004 when technology gurus said innovation in Java is gone down and Sun Microsystems [Now Or...

• **Design Pattern in Java : Part 1**
'Java investment bank interview' generally contains 'Java Design Pattern' questions. If you want to be a professional Java ...

Posted by [Java Interview Questions](#)

Recommend this on Google

Labels: [database](#), [top](#)

[Las vegas hotel offers](#)
[Photo](#)
[Operations Management](#)
[Followers](#)
[Online checking accounts](#)
[Data Structure](#)
[Difference](#)
[2 Face](#)
[Application Data](#)
[Application Example](#)
[Las vegas hotel offers](#)
[Photo](#)
[Operations Management](#)
[Followers](#)
[infolinks](#)

7 comments:

- Affity Solutions** 11 September 2014 at 04:41
Thanks for your ideas. You can also find the details on Affity Solutions, at the [Java Developers](#). The main object of the Affity Solutions is to provide quality web services and is among the few software development company in Nagpur.
[Reply](#)
- Aiswarya Mc** 27 April 2015 at 04:04
IT interview questions and answers for freshers look this page
<http://askxpert.com/>
[Reply](#)
- Kix Mr** 29 October 2015 at 23:19
Tks very much for your post.

Avoid surprises — interviews need preparation. Some questions come up time and time again — usually about you, your experience and the job itself. We've gathered together the most common questions so you can get your preparation off to a flying start.

You also find all interview questions at link at the end of this post.

Source: [Top 10 interview questions and answers](#)

Best rgs
[Reply](#)
- Sura Books Center** 6 November 2015 at 01:19
Thanks for sharing this information with us. This is very helpful for me placement training academy
[Reply](#)
- Steve Smith** 25 November 2015 at 03:23
Great and Useful Article.

[Hibernate Training](#)

[Hibernate Online Training](#)

[Hibernate Training Courses](#)

[Hibernate 4 Supported SQL Dialect](#)
[Reply](#)

**Anna** 30 November 2015 at 20:59

Great and Useful Article.

[Online Java Training](#)[Online Java Training India](#)[Java Training](#)[Java Training In Chennai](#)[Java Training Institutes In Chennai](#)[Java360](#)[Reply](#)**Java Proficiency** 4 December 2015 at 09:01Very Nice ... visit [more java interview questions](#)[Reply](#)

Enter your comment...

Comment as:

Unknown (Goc ▼)

[Publish](#)[Preview](#)[Sign out](#)☐ [Notify me](#)[Newer Post](#)[Home](#)[Older Post](#)[Subscribe to: Post Comments \(Atom\)](#)

INFOLINKS

BEST OFFERS



Best Java Jobs

[Java Software Architect](#)

Oxford, Oxfordshire, United Kingdom

ReThink Recruitment

[Mobile App\(Android\) Automation Engineer](#)

Phoenix, AZ

USG

[Senior C++/Java Developer \(Dublin City\)](#)

Dublin, Dublin, Ireland

Harvey Nash Group

[Java JavaScript Full-Stack Web...](#)

Dublin, Dublin, Ireland

CPL Solutions

[Senior Software Engineer - J2EE...](#)

Hoofddorp, Noord-Holland, Netherlands

Octagon

[Backend Developer](#)

Mumbai, Maharashtra, India

retailending.com

[Long Term TIBCO Developer Contract](#)

Den Haag, Zuid-Holland, Netherlands

Gazelle Global Consulting

[Java Developer - Fluent Dutch](#)

Amsterdam, Noord-Holland, Netherlands

Next Ventures Ltd

[Senior Software Development Engineer](#)

Walnut Creek, CA

Exadel

[Senior Programmer Analyst](#)

Plano, TX





The Irwin Law Firm

[POST A JOB >](#)

POWERED BY JOBTREAD



MUST READ JAVA BOOK

 <p>Effective Java (English) 2nd ...</p> <p>List Price: Rs.699 Our Price: Rs.559</p> <p>Buy from </p>	 <p>Java Concurrency in Practice ...</p> <p>List Price: Rs.425 Our Price: Rs.425</p> <p>Buy from </p>
--	--

POPULAR POSTS

Unix commands : Top 10 frequently asked interview questions

Java Memory Management

Hibernate Interview questions

Java Interviews Frequently Asked Puzzles

Data Structure questions for Java developer

Database Interview Questions for Java Developers

Java 1.7 features

JMS Interview questions

Design Pattern in Java : Part 1

java interview with Investment bank -4

FACEBOOK

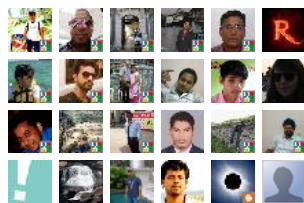
GOOGLE+ FOLLOWERS

FOLLOWERS

Join this site



with Google Friend Connect

Members (40) [More »](#)Already a member? [Sign in](#)

TOTAL PAGEVIEWS

**733,552**

ABOUT ME

**Java Interview Questions**

IT Designer/Architect in Investment Bank with more than 12+ years of experience in design and development of banking java applications.

[View my complete profile](#)**Jobs from Indeed**[Java Developer](#)

Happiest Minds - Chennai, Tamil Nadu

[Java Developer](#)

AEQUALISYS PVT LTD - Chennai, Tamil Nadu

[Java Developer](#)

GAC Pvt Ltd - Chennai, Tamil Nadu

[Software Developer Trainee](#)

Solvosoft - Chennai, Tamil Nadu

[Core Java Developers_Trivandrum](#)

Triadss Tech Solutions PVT Ltd - Chennai, Tamil Nadu

[Java Developer](#)

Innoppl Technologies - Chennai, Tamil Nadu

[Freshers wanted Software developer...](#)

IT Companies - Chennai, Tamil Nadu

[Software Engineer](#)

3Edge Solutions Pvt Ltd - Chennai, Tamil Nadu

[Java & android developers](#)

sortkeys software solutions - Chennai, Tamil Nadu

[JAVA DEVELOPER](#)

Cynosure - Chennai, Tamil Nadu

[1](#) [2](#) [3](#) [4](#) [Next »](#)What: Where:

GOOGLE ANALYTICS

Picture Window template. Template images by Bim. Powered by Blogger.