

Industrial strength Java/JEE Career Companion to open more doors

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) › [member-paid](#) › ♦ 14+ SQL interview Questions & Answers

♦ 14+ SQL interview Questions & Answers

Posted on [August 19, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓

0
Like
Share

Tweet

0
G+1
Share

Q1. Can you explain DDL statements in regards to SQL?

A1. DDL stands for Data Definition Language, which are statements used to define the database structure or schema.

CREATE – to create objects in the database, **ALTER** – alters the structure of the database, **DROP** – delete objects from the database, **TRUNCATE** – remove all records from a table, including all spaces allocated for the records are removed, **COMMENT** – add comments to the data dictionary, and **RENAME** – rename an object.

Create a table named “EMPLOYEES” using a DDL statement

```
1 CREATE TABLE "EMPLOYEES"
2   (   "ID" NUMBER(*,0) NOT NULL ENABLE,
3     "CODE" NUMBER(*,0) NOT NULL ENABLE,
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

✚ Ice Breaker Interview

✚ Core Java Interview C

✚ JEE Interview Q&A (3

✚ Pressed for time? Jav

✚ SQL, XML, UML, JSC

✚ ERD (1)

✚ NoSQL (2)

✚ Regex (2)

✚ SQL (7)

 ♦ 15 Database d

 ♦ 14+ SQL interv

 ♦ 9 SQL scenari

 Auditing databas

 Deleting records

 SQL Subquery ir

 Transaction man

✚ UML (1)

✚ JSON (2)

✚ XML (2)

✚ XSD (2)

✚ YAML (2)

```

4  "NAME" VARCHAR2(1024 BYTE),
5  CONSTRAINT "EMPLOYEES_PK" PRIMARY KEY ("ID")
6  CONSTRAINT "EMPLOYEES_UQ" UNIQUE ("CODE"));
7

```

Create a table named "EXECUTIVES" using a DDL statement

```

1 CREATE TABLE "EXECUTIVES"
2 (
3   "ID" NUMBER(*,0) NOT NULL ENABLE,
4   "CODE" NUMBER(*,0) NOT NULL ENABLE,
5   "AREA" VARCHAR2(1024 BYTE),
6   CONSTRAINT "EXECUTIVES_PK" PRIMARY KEY ("ID")
7   CONSTRAINT "EXECUTIVES_FK" FOREIGN KEY ("CODE")
8   CONSTRAINT "EXECUTIVES_UQ" UNIQUE ("CODE"));

```

```

1 CREATE SEQUENCE "EMPLOYEES_SEQ" MINVALUE 1 MAXV
2 CREATE SEQUENCE "EXECUTIVES_SEQ" MINVALUE 1 MAX
3

```

Q2. Can you explain DML statements in regards to SQL?

A2. DML stands for Data Manipulation Language, which are used for managing data within schema objects. **SELECT** – retrieve data from the a database, **INSERT** – insert data into a table, **UPDATE** – updates existing data within a table, **DELETE** – deletes all records from a table, the space for the records remain, **CALL** – call a PL/SQL or Java subprogram, **EXPLAIN PLAN** – explain access path to data, and **LOCK TABLE** – control concurrency are DML statements.

To insert data into abbe tables:

```

1 INSERT INTO EMPLOYEES(ID, CODE, NAME ) VALUES (EM
2 INSERT INTO EMPLOYEES(ID, CODE, NAME ) VALUES (EM
3 INSERT INTO EMPLOYEES(ID, CODE, NAME ) VALUES (EM
4 INSERT INTO EMPLOYEES(ID, CODE, NAME ) VALUES (EM
5 INSERT INTO EMPLOYEES(ID, CODE, NAME ) VALUES (EM
6
7 INSERT INTO "EXECUTIVES" ("ID", "CODE", "AREA" )
8 INSERT INTO "EXECUTIVES" ("ID", "CODE", "AREA" )
9

```

Also, note that **DCL** stands for Data Control Language (DCL) statements with **GRANT** – gives user's access privileges to database and **REVOKE** – withdraw access privileges given with the GRANT command, and **TCL** statements stand for Transaction Control (TCL) with statements are **COMMIT** –

- [Hadoop & BigData Int](#)
- [Java Architecture Inte](#)
- [Scala Interview Q&As](#)
- [Spring, Hibernate, & I](#)
- [Testing & Profiling/Sa](#)
- [Other Interview Q&A 1](#)
- [Free Java Interview](#)

16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience I](#)
- [Low Latency \(7\)](#)
- [Memory Managemen](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

80+ step by step Java Tutorials

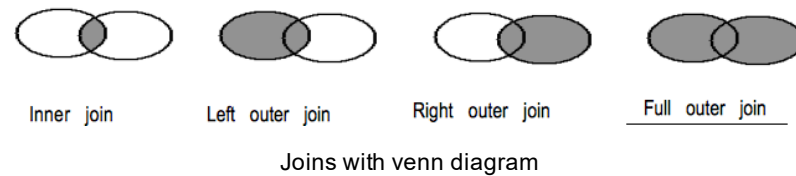
[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2](#)
- [Hadoop & Spark Tuto](#)

save work done, **SAVEPOINT** – identify a point in a transaction to which you can later roll back, **ROLLBACK** – restore database to original since the last COMMIT, and **SET TRANSACTION** – Change transaction options like isolation level and what rollback segment to use.

Q3. Explain inner and outer joins? ?

A3. Joins allow database users to combine data from one table with data from one or more other tables (or views, or synonyms). Tables are joined two at a time making a new table containing all possible combinations of rows from the original two tables.



ID	CODE	NAME
1	10	JOHN
2	20	Sam
3	30	Peter
4	40	Dan
5	50	Emma

Employees Table

ID	CODE	AREA
7	20	REGION1
8	50	REGION2

Executives Table

Inner join

```

1 select em.*, ex.id as exec_id, ex.code as exec_co
2   where em.code = ex.code;
3
4 or
5
6 select em.*, ex.id as exec_id, ex.code as exec_co

```

- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorials \(19\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [Complete the given code \(1\)](#)
- [Converting from A to B \(1\)](#)
- [Designing your class \(1\)](#)
- [Java Data Structures \(1\)](#)
- [Passing the unit tests \(1\)](#)
- [What is wrong with this code? \(1\)](#)
- [Writing Code Home A \(1\)](#)
- [Written Test Core Java \(1\)](#)
- [Written Test JEE \(1\)](#)

How good are your?

[open all](#) | [close all](#)

- [Career Making Knowledge \(1\)](#)
- [Job Hunting & Resumes \(1\)](#)

```

7 inner join Executives ex on em.code = ex.code;
8
9

```

ID	CODE	NAME	EXEC_ID	EXEC_CODE	AREA
2	20	Sam	7	20	REGION1
5	50	Emma	8	50	REGION2

SQL inner join

Left outer join

```

1 select em.*, ex.id as exec_id, ex.code as exec_co
2 left outer join Executives ex on em.code = ex.co
3

```

ID	CODE	NAME	EXEC_ID	EXEC_CODE	AREA
1	10	JOHN	(null)	(null)	(null)
2	20	Sam	7	20	REGION1
3	30	Peter	(null)	(null)	(null)
4	40	Dan	(null)	(null)	(null)
5	50	Emma	8	50	REGION2

SQL left outer join

Right outer join

```

1 select em.*, ex.id as exec_id, ex.code as exec_c
2 right outer join Executives ex on em.code = ex
3

```

ID	CODE	NAME	EXEC_ID	EXEC_CODE	AREA
2	20	Sam	7	20	REGION1
5	50	Emma	8	50	REGION2

SQL right outer join

Full outer join

```

1 select em.*, ex.id as exec_id, ex.code as exec
2 full outer join Executives ex on em.code = ex.
3

```

R2	ID	R2	CODE	R2	NAME	R2	EXEC_ID	R2	EXEC_CODE	R2	AREA
	1		10		JOHN		(null)		(null)		(null)
	2		20		Sam		7		20		REGION1
	3		30		Peter		(null)		(null)		(null)
	4		40		Dan		(null)		(null)		(null)
	5		50		Emma		8		50		REGION2

SQL full outer join

Q4. What is a self join?

A4. A self-join is a join of a table to itself. In certain scenarios, a self join is a better alternative to a sub-query.

Step 1: If you want to add a new column called “Manager” to “Employees” table, you can use the ALTER DDL.

```
1 ALTER table EMPLOYEES add (MANAGER_CODE NUMBER(*,
2
```

Step 2: To update who the manager is with the following DML

```
1 UPDATE EMPLOYEES set MANAGER_CODE=20 where CODE I
2 UPDATE EMPLOYEES set MANAGER_CODE=50 where CODE I
3
```

R2	ID	R2	CODE	R2	NAME	R2	MANAGER_CODE
	1		10		JOHN		20
	2		20		Sam		(null)
	3		30		Peter		50
	4		40		Dan		20
	5		50		Emma		(null)

SQL all employees with manager_code

Step 3: SQL self join can be used to print employee name and manager name side by side. Basically, you are joining the same table to each other.

EMPLOYEE	MANAGER
Dan	Sam
JOHN	Sam
Peter	Emma

SQL self join

Q5. What is a sub-query? How does a sub-query impact on performance? What are the advantages and disadvantages of sub-queries?

A5. It is possible to embed a SQL statement within another. When this is done on the WHERE or the HAVING statements, we have a subquery construct. What is subquery useful for? It is used to join tables and there are cases where the only way to correlate two tables is through a sub-query.

```
1 SELECT emp.name FROM employees emp WHERE
2 emp.code NOT IN (SELECT code FROM executives);
3
```

There can be performance problems with sub-queries. The above query can be re-written as an **outer join** for a faster performance as shown below:

```
1 SELECT emp.name FROM employees emp left outer joi
2 on emp.code = exec.code;
3
```

Advantages:

- 1) Sub-queries allow you to use the results of another query in the outer query.
- 2) Sub-queries in some complex SQL queries can simplify coding and improve maintainability by breaking down the complex query into a series of logical steps.
- 3) In some cases, subqueries are easier to understand than complex joins and unions.

Disadvantages:

When a sub-query is used, the query optimizer of the database server may have to perform additional steps like sorting the results, etc. Hence, in some cases sub-queries can be less efficient than using joins.

Q6. What is a correlated sub-query?

A6. A query is called correlated sub-query when both the inner query and the outer query are interdependent. For every row processed by the inner query, the outer query is processed as well. The inner query depends on the outer query before it can be processed.

```
1 SELECT outer.product_name FROM product outer
2 WHERE outer.product_id = (SELECT inner.product_i
3                           WHERE outer.product_i
4
```

If a subquery is not dependent on the outer query it is called a **non-correlated subquery**.

Q7. Can you explain aggregate SQL functions?

A7. SQL provides aggregate functions to assist with the summarization of large volumes of data.

ORDERID	FIRSTNAME	SURNAME	QTY	UNITPRICE
1001	John	Darcy	25	10.5
1002	Peter	Smith	25	10.5
1003	Sam	Greg	25	10.5

SQL aggregate

```
1 SELECT SUM(QTY) AS Total FROM Orders; -- Total =
2 SELECT AVG(UnitPrice * QTY) As AveragePrice FROM
3
```

If we inserted another row to the above table:

ORDERID	FIRSTNAME	SURNAME	QTY	UNITPRICE
1004	John	Darcy	20	10.50

SQL aggregate

```
1 SELECT FIRSTNAME, SUM(QTY) FROM orders
2   GROUP BY FIRSTNAME
3   HAVING SUM(QTY) > 25; -- John 45
4
```

Q8. How can you compare a part of the name rather than the entire name?

A8. You can use wild card characters like:

* (% in oracle) : Matches any number of characters.

? (_ in oracle) : Matches a single character.

To find all the employees who have “au” in their names

```
1 SELECT * FROM employees emp
2   WHERE emp.name LIKE '%au%';
3
```

Q9. How do you get distinct entries from a table?

A9. The SELECT statement in conjunction with DISTINCT lets you select a set of distinct values from a table.

```
1 SELECT DISTINCT name FROM employees;
```

Q10. How can you find the total number of records in a table?

A10. Use the **COUNT** key word:

```
1 SELECT COUNT(*) FROM employees WHERE emp.name LI
```

Q11. In your experience, what are some of the common mistakes developers make related to SQL?

A11.

1. Cartesian joins

SQL Joins are used to relate information in different tables. A Join condition is a part of the sql query that retrieves rows from two or more tables. A SQL Join condition is used in the

SQL WHERE Clause of select, update, delete statements. If a sql join condition is omitted as shown below:

```
1 SELECT col1, col2, col3
2 FROM table_name1, table_name2
```

or if the condition is invalid, then the join operation will result in a Cartesian product. The Cartesian product returns a number of rows equal to the product of all rows in all the tables being joined. For example, if the first table has 20 rows and the second table has 10 rows, the result will be $20 * 10$, or 200 rows. This query will take a long time to execute.

2. Use of SELECT *

For example, a common misuse of SELECT * is to extract a set of all employees and to insert them into another table called Contractors with the same structure

```
1 INSERT INTO Contractors
2   SELECT * FROM Employees WHERE emp_type = 'C';
3
```

The above query does the job, however, one day business requirements change and two new columns are added to the Employees table:

```
1 ALTER TABLE Products
2   ADD effective_start_date DATETIME, effective_en
3
```

All of sudden the query that extracts from the Employees table and insert records into the Contractor table results in error.

“Insert Error: Column name or number of supplied values does not match table definition.”

The fix is to explicitly list the column names in the query:

```
1 INSERT INTO Contractors (emp_id, emp_name)
2   SELECT emp_id, emp_name FROM Employees WHERE em
3
```

3. Not using Prepared statements. Prepared statements are more secured and efficient than the ordinary statements. Prepared statements prevent SQL injection attacks.

4. Using the predicate “LIKE” in indexed columns. The “LIKE” predicate typically performs a search without the normal performance benefit of indexes. Using ‘=’, ‘<>’, etc instead of “LIKE” will increase performance. Also should be aware of that case sensitivity (e.g., ‘A’ versus ‘a’) may be different based upon database Server or configuration.

5. Over use of cursors in stored procedures. If possible, avoid using SQL stored proc cursors. They generally use a lot of Server resources and reduce the performance and scalability of your applications. If you need to perform row-by-row operations, try to find another method to perform the task.

Q12. What is the difference between “Truncate” and “Delete” commands?

A12. TRUNCATE is useful for purging a table with huge amount of data. Alternatively, you can drop the table and recreate it that makes sense. Firing a delete command instead of a truncate command to empty a table with millions of records can result in locking the whole table and also can take longer time to complete, and at times cause the machine to hang.

```
1 TRUNCATE TABLE table_name
```

a) TRUNCATE TABLE_NAME always locks the table and page but not each row, whereas DELETE statement is executed using a row lock, each row in the table is locked for deletion.

b) Truncate removes all the records in the table whereas delete can be used with WHERE clause to remove records conditionally. That is remove only a handful number of

records.

c) Truncate performance is much faster than Delete, as its logging is minimal whereas the Delete command logs every record.

d) Truncate does not retain the identity, whereas DELETE command retains the identity. When you use Truncate, If the table contains an identity column, the counter for that column is reset to the seed value that is defined for the column.

e) Truncate cleans up the object statistics and clears the allocated space whereas Delete retains the object statistics and allocated space.

f) TRUNCATE is a DDL (Data Definition Language) and DELETE is a DML (Data Manipulation Language).

g) Data removed by TRUNCATE command cannot be generally rolled back unless the database server specifically supports it. The DELETE command can rollback a transaction.

h) The TRUNCATE command does not fire any triggers, whereas the DELETE command fires any triggers defined on the table. For example, to keep an audit trail of records that have been deleted by inserting the deleted records into an audit table via the DELETE triggers.

Q. Which command will you use to periodically purge data from your tables as part of a house keeping job?

A. Use a DELETE command within a transaction with a WHERE clause to remove data that are older than 7 years. Remove large amount of data in batches as opposed to in a single transaction.

Q13. How will you go about deleting a few records from parent and child tables where the parent table with parent_name = 'Peter'?

A13. Firstly, you need to delete the child records because the integrity constraint won't let you delete the parent record when there are child records.

```
1 DELETE child FROM parent p, child c
2 WHERE p.parent_id = c.parent_id AND p.parent_name = 'Peter'
3
```

Now, the parent table can be deleted as shown below

```
1 DELETE FROM parent p WHERE p.parent_name = 'Pete'
```

Q14. What do you do with the PURGE command?

A14. The **purge** command is used to clear the recycle bin. It is generally used with the DROP command. For example,

```
1 drop table tablename purge;
```

the above command will clear away the table from database as well as from the recycle bin. After executing the purge command, you cannot retrieve the table using a flashback query.

Q15. How will you improve the performance of the following SQL query?

```
1  
2 SELECT * FROM Customers  
3     WHERE CustomerName NOT IN ('John', 'Pet  
4
```

A15. Pick only the columns you are interested in and replace “NOT IN” with “<> AND <>” as shown below.

```
1  
2 SELECT ContactName, Address, City FROM Customers  
3     WHERE CustomerName <> 'John' and Custome  
4
```

Popular Posts

♦ [11 Spring boot interview questions & answers](#)

828 views

♦ [Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers](#)

768 views

[18 Java scenarios based interview Questions and Answers](#)

400 views

001A: ♦ 7+ Java integration styles & patterns
interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions
& answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time
in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview
Questions and Answers

280 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces
and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts
interview questions & answers

202 views

Bio

Latest Posts



Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



About [Arulkumaran Kumaraswamipillai](#)



Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ ♦ 15 Database design interview Questions & Answers

01: ♥♦ 40+ Java Web Services Architecture & Basics Interview Q&A –

Q01 – Q12 ▶

Posted in member-paid, SQL

Tags: Java/JEE FAQs, JEE FAQs, Novice FAQs

Leave a Reply

Logged in as geethika. [Log out?](#)

Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)
☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.