**SQL Interview question**

**Stored procedure**

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored proc

edure, so that the stored procedure can act based on the parameter value(s) that is passed.

### Stored Procedure Syntax

CREATE PROCEDURE procedure\_name  
AS  
sql\_statement  
GO;

### Execute a Stored Procedure

EXEC procedure\_name;

A **trigger** is a special type of stored procedure that automatically runs when an event occurs in the database server. DML **triggers** run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]

**Explanation of syntax:**

1. create trigger [trigger\_name]: Creates or replaces an existing trigger with the trigger\_name.
2. [before | after]: This specifies when the trigger will be executed.
3. {insert | update | delete}: This specifies the DML operation.
4. on [table\_name]: This specifies the name of the table associated with the trigger.
5. [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
6. [trigger\_body]: This provides the operation to be performed as trigger is fired

**BEFORE and AFTER of Trigger:**  
BEFORE triggers run the trigger action before the triggering statement is run.  
AFTER triggers run the trigger action after the triggering statement is run.

**Example:**  
Given Student Report Database, in which student marks assessment is recorded. In such schema, create a trigger so that the total and average of specified marks is automatically inserted whenever a record is insert.

Here, as trigger will invoke before record is inserted so, BEFORE Tag can be used.

create trigger stud\_marks

before INSERT

on

Student

for each row

set Student.total = Student.subj1 + Student.subj2 + Student.subj3, Student.per = Student.total \* 60 / 100;

SELECT name, salary FROM #Employee e1 WHERE N-1 = (SELECT COUNT(DISTINCT salary) FROM #Employee e2 WHERE e2.salary > e1.salary)

SELECT TOP 1 salary

FROM (

SELECT DISTINCT TOP N salary

FROM #Employee

ORDER BY salary DESC

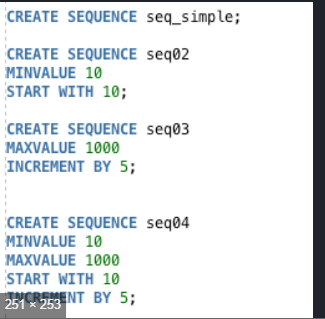
) AS temp

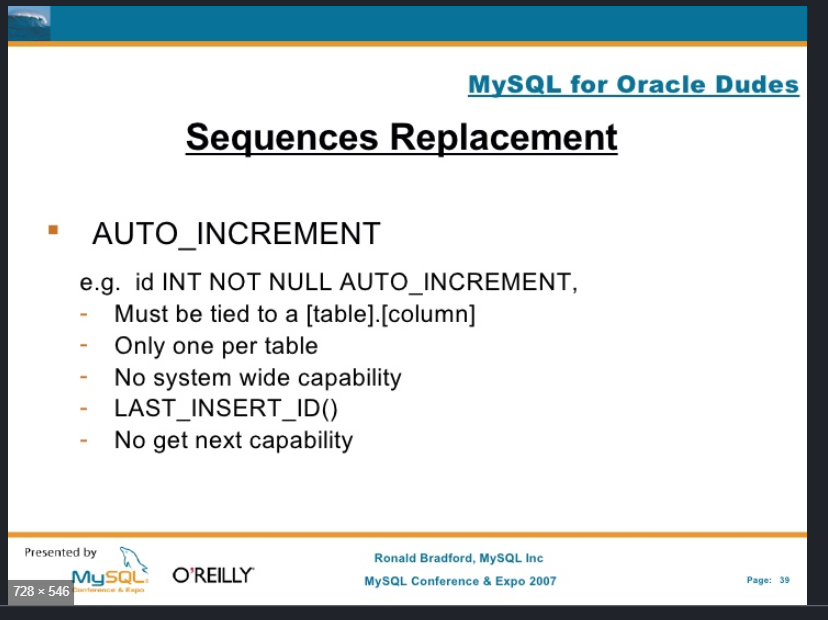
ORDER BY salary

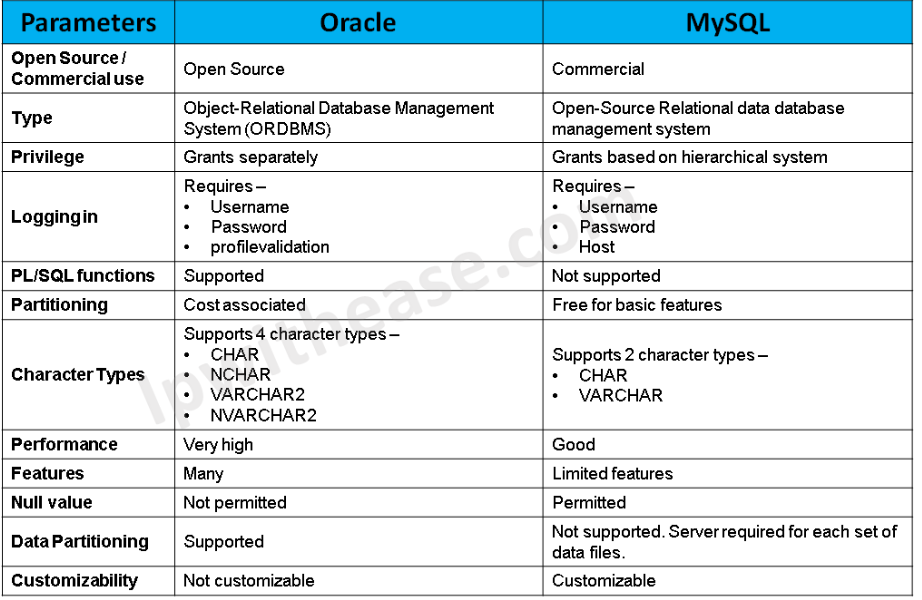
SELECT salary FROM Employee ORDER BY salary DESC LIMIT N-1, 1

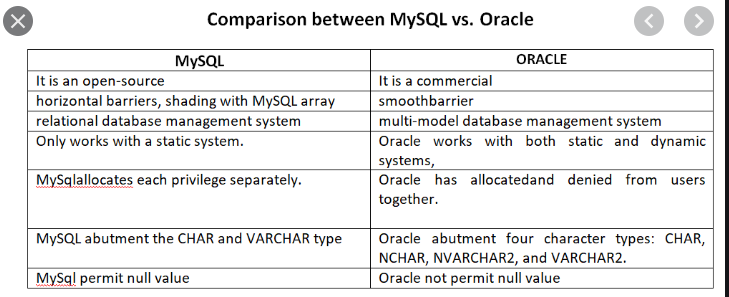
<https://howtodoinjava.com/sql/sql-query-find-nth-highest-salary/>

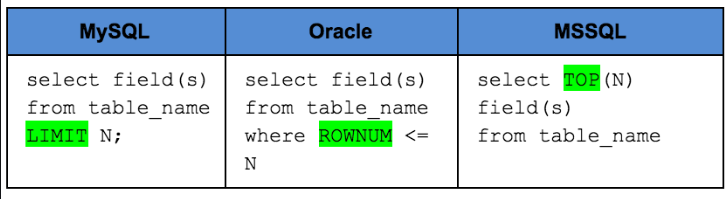
A **sequence** is a distinct database object in **Oracle**. In **MySQL**, when you have an **autoincrement** column and you INSERT a new row in a table, you simply don't mention the **autoincrement** column and **MySQL** puts it there. You can then insert the same number into another table by referencing LAST\_INSERT\_ID()

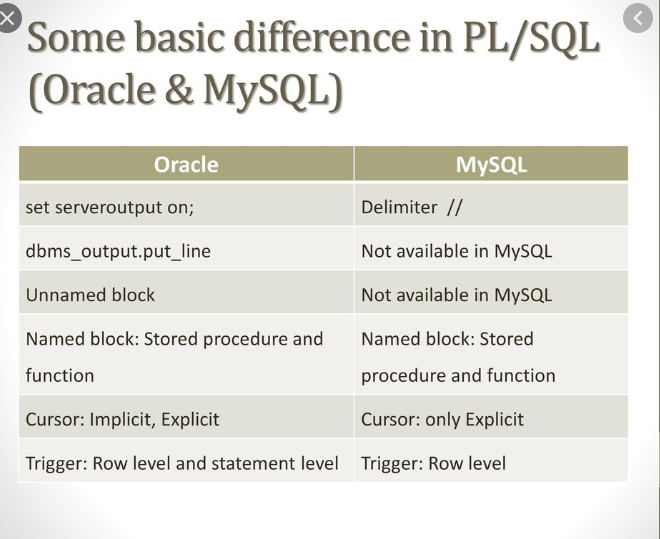












1. **Name the different types of indexes in SQL and define them.**

**Unique Index:** Prevents duplicate entries within uniquely indexed columns. They are automatically generated if a Primary Key is available.

**Clustered Index:** Used to organize or edit the arrangement within the table, with respect to the key value. Each table is only allowed to have a single clustered index only.

**NonClustered Index:** Conversely, NonClustered Index only manages the order of logic within entries. It does not manage the arrangement and tables can have multiple NonClustered Indexes.

### How to select 10 records from a table?

MySQL: Using limit clause, example select \* from Employee limit 10;

Oracle: Using ROWNUM clause, example SELECT \* FROM Employee WHERE ROWNUM < 10;

SQL Server: Using TOP clause, example SELECT TOP 3 \* FROM Employee;

1. **hat is a trigger in SQL?**

A database trigger is a program that automatically executes in response to some event on a table or view such as insert/update/delete of a record. Mainly, the database trigger helps us to maintain the integrity of the database.

1. **What is Auto Increment feature in SQL?**

Auto increment allows the user to create a unique number to be generated whenever a new record is inserted in the table. AUTO INCREMENT is the keyword for Oracle, AUTO\_INCREMENT in MySQL and IDENTITY keyword can be used in SQL SERVER for auto-incrementing. Mostly this keyword is used to create the primary key for the table.

1. **What is collation?**

Collation is basically a set of rules on how to compare and sort characters, extended to strings. Collation in MSSQL and MySQL works pretty much the same way, except on certain collation options such as UTF-8. Besides the normal character-wise comparison, collation can also sort and compare strings on an ASCII representation perspective.

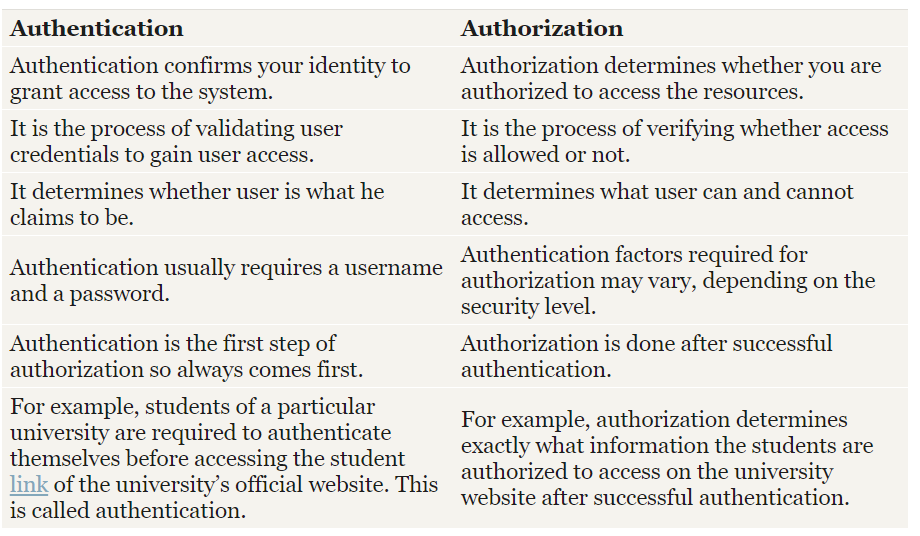
1. **What is a recursive stored procedure?**

A stored procedure which calls by itself until it reaches some boundary condition. This recursive function or procedure helps programmers to use the same set of code any number of times.

<https://www.journaldev.com/17773/sql-interview-questions-answers#what-is-auto-increment-feature-in-sql>

What is authentication and authorization?

Difference between **Authentication and Authorization**. ... **Authentication** means confirming your own identity, while **authorization** means granting access to the system. In simple terms, **authentication** is the process of verifying who you are, while **authorization** is the process of verifying what you have access to.



Based on the security level, authentication factor can vary from one of the following:

* **Single-Factor** **Authentication** – It’s the simplest authentication method which commonly relies on a simple password to grant user access to a particular system such as a [website](http://www.differencebetween.net/technology/difference-between-landing-page-and-website/) or a network. The person can request access to the system using only one of the credentials to verify his identity. The most common example of a single-factor authentication would be login credentials which only require a password against a username.
* **Two-Factor Authentication**– As the name suggests, it’s a two-step verification process which not only requires a username and password, but also something only the user knows, to ensure an additional level of security, such as an ATM pin, which only the user knows. Using a username and password along with an additional piece of confidential information makes it virtually impossible for fraudsters to steal valuable data.
* **Multi-Factor Authentication**– It’s the most advanced method of authentication which uses two or more levels of security from independent categories of authentication to grant user access to the system. All the factors should be independent of each other to eliminate any vulnerability in the system. Financial organizations, banks, and law enforcement agencies use multiple-factor authentication to safeguard their data and applications from potential threats.

<http://www.differencebetween.net/technology/difference-between-authentication-and-authorization/>

Devops interview question

**Which are the top DevOps tools? Which tools have you worked on?**

The most popular DevOps tools are mentioned below:

* Git : Version Control System tool
* Jenkins : Continuous Integration tool
* Selenium : Continuous Testing tool
* Puppet, Chef, Ansible : Configuration Management and Deployment tools
* Nagios : Continuous Monitoring tool
* Docker : Containerization tool

**Q5. How do all these tools work together?**

Given below is a generic logical flow where everything gets automated for seamless delivery. However, this flow may vary from organization to organization as per the requirement.

1. Developers develop the code and this source code is managed by Version Control System tools like Git etc.
2. Developers send this code to the Git repository and any changes made in the code is committed to this Repository.
3. Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.
4. Configuration management tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
5. Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).
6. After deployment It is continuously monitored by tools like Nagios.
7. Docker containers provides testing environment to test the build features.

**What is Version control?**

This is probably the easiest question you will face in the interview. My suggestion is to first give a definition of Version control. It is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control systems consist of a central shared repository where teammates can commit changes to a file or set of file. Then you can mention the uses of version control.

Version control allows you to:

* Revert files back to a previous state.
* Revert the entire project back to a previous state.
* Compare changes over time.
* See who last modified something that might be causing a problem.
* Who introduced an issue and when.

### Q5. What is Git?

I will suggest that you attempt this question by first explaining about the architecture of git as shown in the below diagram. You can refer to the explanation given below:

* Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.
* Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage is that it does not rely on a central server to store all the versions of a project’s files. Instead, every developer “clones” a copy of a repository I have shown in the diagram below with “Local repository” and has the full history of the project on his hard drive so that when there is a server outage, all you need for recovery is one of your teammate’s local Git repository.
* There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are commiting changes “Remote repository”.

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages.

