<https://beginnersbook.com/2015/04/constraints-in-dbms/>

**Java CallableStatement Interface:**

CallableStatement interface is used to call the **stored procedures and functions**.

We can have business logic on the database by the use of stored procedures and functions that will make the performance better because these are precompiled.

Suppose you need the get the age of the employee based on the date of birth, you may create a function that receives date as the input and returns age of the employee as the output.

==========================

DQL (Data Query Language): Select

DDL(Data Definition Language): create ,Drop, Alter,truncate, comment, Rename

DML(Data Manupulation Language): Select , Insert, Update, Delete. Lock, Call , Explain Plan

DCL(Data Control Language): GRANT, Revoke

# ---------------------------------------------------------------------------------------------------------

# What is the difference between stored procedures and functions.

|  |  |
| --- | --- |
| **Stored Procedure** | **Function** |
| is used to perform business logic.  must not have the return type.  may return 0 or more values.  We can call functions from the procedure.  Procedure supports input and output parameters.  Exception handling using try/catch block can be used in stored procedures. | is used to perform calculation.  must have the return type.  Procedure cannot be called from function.  may return only one values.  Function supports only input parameter.  Exception handling using try/catch can't be used in user defined functions. |

**---------------------------------------------------------------------------------------------------------------**

# Prepared Statement Interface

The Prepared Statement interface is a sub-interface of Statement. It is used to execute parameterized query.

### Why use PreparedStatement?

**Improves performance**: The performance of the application will be faster if you use PreparedStatement interface because query is compiled only once.

**----------------------------------------------------------------------------------------------------------**

# Statement interface

The **Statement interface** provides methods to execute queries with the database. The statement interface is a factory of ResultSet i.e. it provides factory method to get the object of ResultSet.

### Commonly used methods of Statement interface:

The important methods of Statement interface are as follows:

* **public ResultSet executeQuery(String sql):** is used to execute SELECT query. It returns the object of ResultSet.
* **2) public int executeUpdate(String sql):** is used to execute specified query, it may be create, drop, insert, update, delete etc.
* **3) public boolean execute(String sql):** is used to execute queries that may return multiple results.
* **4) public int[] executeBatch():** is used to execute batch of commands.

## Top Answers to MySQL Interview Questions

### ****1. What do DDL, DML, and DCL stand for?****

DDL is the abbreviation for Data Definition Language dealing with database schemas, as well as the description of how data resides in the database. An example of this is the CREATE TABLE command.

DML denotes Data Manipulation Language which includes commands such as SELECT, INSERT, etc.

DCL stands for Data Control Language and includes commands like GRANT, REVOKE, etc.

### ****2. Compare MySQL vs SQL Server.****

|  |  |  |
| --- | --- | --- |
| **Criteria** | **MySQL** | **SQL Server** |
| Developed by | Oracle | Microsoft |
| Programmed in | C and C++ | Mainly C++, but some parts in C |
| Platforms | Supports many platforms | Supports only Linux and Windows |
| Syntax | Complex Syntax | Simpler and easy-to-use syntax |

### ****2. What is SQL Server?****

SQL Server is one of the database management systems (DBMS) and is designed by Microsoft.  DBMS are computer software applications with the capability of interacting with users, various other applications, and databases. The objective of SQL Server is capturing and analyzing data and managing the definition, querying, creation, updating, and administration of the database.

## ****Watch this SQL Tutorial for Beginners video:****

Learn for free ! Subscribe to our youtube Channel.

### ****3. How and why use SQL Server?****

SQL Server is free and anyone can download and use it. The application uses SQL (Structured Query Language), and it is easy to use.

### ****4. What are the features of MySQL?****

MySQL provides cross-platform support, a wide range of interfaces for application programming, and has many stored procedures like triggers and cursors that help in managing the database.

### ****5. What is the Traditional Network Library for a system?****

In either Windows or POSIX systems, the named pipes provide ways of inter-process communications to connect different processes running on the same machine. It dispenses with the necessity of using the network stack, and data can be sent without affecting the performance. Servers set up named pipes to listen to requests. The client process needs to know the specific pipe name to send the request.

### ****6. What is the default port for MySQL Server?****

The default port for MySQL Server is 3306. Another standard default port is 1433 in TCP/IP for SQL Server.

***Learn all about SQL through this***[***SQL Certification Course***](https://intellipaat.com/microsoft-sql-server-certification-training/)***!***

***7*. What is a join in MySQL?**

In MySQL, joins are used to query data from two or more tables. The query is made using the relationship between certain columns existing in the table. There are four types of joins in MySQL. Inner join returns rows if there is at least one match in both tables. Left join returns all the rows from the left table even if there is no match in the right table. Right join returns all the rows from the right table even if no matches exist in the left table. Full join would return rows when there is at least one match in the tables.

**Master SQL concepts in depth from this**[***SQL Tutorial***](https://intellipaat.com/blog/tutorial/sql-tutorial/)**now!**

### ****9. What are the common MySQL functions?****

Common MySQL functions are as follows:

* **NOW():** The function for returning the current date and time as a single value
* **CURRDATE():** The function for returning the current date or time
* **CONCAT (X, Y):** The function to concatenate two string values creating a single string output
* **DATEDIFF (X, Y):** The function to determine the difference between two dates

***Check out Intellipaat’s blog to get a fair understanding of***[***SQL Optimization Techniques***](https://intellipaat.com/blog/sql-optimization-techniques/)***!***

### ****10. What is the difference between CHAR and VARCHAR?****

When a table is created, CHAR is used to define the fixed length of the table and columns. The length value could be in the range of 1–255. The VARCHAR command is used to adjust the column and table lengths as required.

### ****11. What are Heap Tables?****

Basically, Heap tables are in-memory tables used for high-speed temporary storage. But, TEXT or BLOB fields are not allowed within them. They also do not support AUTO INCREMENT.

### ****12. What is the syntax for concatenating tables in MySQL?****

The syntax for concatenating tables is MySQL:

CONCAT (string 1, string 2, string 3)

### ****13. What is the limit of indexed columns that can be created for a table?****

The maximum limit of indexed columns that can be created for any table is 16.

### ****14. What are the different types of strings used in database columns in MySQL?****

In MySQL, the different types of strings that can be used for database columns are SET, BLOB, VARCHAR, TEXT, ENUM, and CHAR.

### ****15. How can a user get the current SQL version?****

The syntax for getting the current version of MySQL:

SELECT VERSION ();

### ****16. Is there an object-oriented version of MySQL library functions?****

Yes. MySQLi is the object-oriented version of MySQL, and it interfaces in PHP.

### ****17. What is the storage engine used for MySQL?****

Storage tables are named as table types. The data is stored in the files using multiple techniques such as indexing, locking levels, capabilities, and functions.

### ****18. What is the difference between the primary key and the candidate key?****

The primary key in MySQL is used to identify every row of a table in a unique manner. For one table, there is only one primary key. The candidate keys can be used to reference the foreign keys. One of the candidate keys is the primary key.

**Want to get certified in SQL? Go through our blog on**[***SQL Server Certification***](https://intellipaat.com/blog/sql-server-certification/)**and be informed!**

### ****19. What are the different types of tables in MySQL?****

MyISAM is the default table that is based on the sequential access method.

* **Heap** is the table that is used for fast data access, but the data will be lost if the table or the system crashes.
* **InnoDB** is the table that supports transactions using the COMMIT and ROLLBACK commands.
* **BDB** can support transactions similar to InnoDB, but the execution is slower.

### ****20. Can you use MySQL with Linux operating system?****

Yes. The syntax for using MySQL with Linux operating system is as follows:

etc/init.d/mysqlstart

### ****21. What is the use of ENUM in MySQL?****

The use of ENUM will limit the values that can go into a table. For instance, a user can create a table giving specific month values and other month values would not enter into the table.

### ****22. What are the TRIGGERS that can be used in MySQL tables?****

Following TRIGGERS are allowed in MySQL:

* BEFORE INSERT
* AFTER INSERT
* BEFORE UPDATE
* AFTER UPDATE
* BEFORE DELETE
* AFTER DELETE

### ****i23. What is the difference between LIKE and REGEXP operators in MySQL?****

* LIKE is denoted using the ‘%’ sign. For example:

SELECT \* FROM user WHERE user name LIKE “%NAME”

* On the other hand, the use of REGEXP is as follows:

SELECT \* FROM user WHERE username REGEXP “^NAME”;

### ****24. How to use the MySQL slow query log?****

Information that is provided on the slow query log could be huge in size. The query could also be listed over a thousand times. In order to summarize the slow query log in an informative manner, one can use the third-party tool ‘pt-query-digest’.

### ****25. How can one take an incremental backup in MySQL?****

A user can take an incremental backup in MySQL using Percona XtraBackup.

### ****26. How can you change the root password if it is lost?****

In such cases when the password is lost, the user should start the DB with skip-grants-table and then change the password. Thereafter, with the new password, the user should restart the DB in a normal mode.

### ****27. How to resolve the problem of the data disk that is full?****

When the data disk is full and overloaded, the way out is to create and soft link and move the .frm and the .idb files into that link location.

### ****28. What is the difference between the DELETE TABLE and TRUNCATE TABLE commands in MySQL?****

Basically, DELETE TABLE is a logged operation, and every row deleted is logged. Therefore, the process is usually slow. TRUNCATE TABLE also deletes rows in a table, but it will not log any of the rows deleted.  The process is faster here in comparison. TRUNCATE TABLE can be rolled back and is functionally similar to the DELETE statement without a WHERE clause.

### ****29. What are the types of joins in MySQL?****

There are four types of joins in MySQL.

Inner join returns the rows if there is at least one match in two tables.

Left join returns all the rows from the left table even if there is no match in the right table.

Right join returns all the rows from the right table even if no matches exist in the left table.

Full join would return rows when there is at least one match in the tables.

### ****30.What are the storage models of OLAP?****

The storage models in OLAP are MOLAP, ROLAP, and HOLAP.

### ****31. How to define the testing of network layers in MySQL?****

For this, it is necessary to review the layered architecture and determine hardware and software configuration dependencies with respect to the application put to test.

### ****32. What is the difference between primary key and unique key?****

While both are used to enforce the uniqueness of the column defined, the primary key would create a clustered index, whereas the unique key would create a non-clustered index on the column. The primary key does not allow ‘NULL’, but the unique key does.

### ****33. What is meant by transaction? What are ACID properties?****

Transaction is a logical unit of work where either all or none of the steps should be performed. ACID is the abbreviation for Atomicity, Consistency, Isolation, and Durability that are properties of any transaction.

### ****34. How can one restart SQL Server in the single user or the minimal configuration modes?****

The command line SQLSERVER.EXE used with ‘–m’ will restart SQL Server in the single user mode and the same with ‘–f’ will restart it in the minimal configuration mode.

### ****35. What is the difference between BLOB and TEXT?****

BLOBs are binary large object holding huge data. Four types of BLOBs are TINYBLOB, BLOB, MEDIBLOB, and LONGBLOB. TEXT is a case-sensitive BLOB. Four types of TEXT are TINY TEXT, TEXT, MEDIUMTEXT, and LONG TEXT.

***If you have any doubts or queries related to SQL, get them clarified from SQL experts on our***[***SQL Community***](https://intellipaat.com/community/sql)***!***

### ****36. What is the basic MySQL architecture?****

The logical architecture of MySQL is made of ‘connection manager’, ‘query optimizer’, and ‘pluggable engines’.

**JoinS**

* **Self join**
* **Inner join**
* **Outer join has 3 type:**
* **1) Left outer join** returns all rows of table on left side of join. The rows for which there is no matching row on right side, result contains NULL in the right side.

SELECT Student.StudentName, StudentCourse.CourseID

FROM Student LEFT OUTER JOIN StudentCourse

ON StudentCourse.EnrollNo = Student.EnrollNo

ORDER BY StudentCourse.CourseID;

**2) Right Outer Join**is similar to Left Outer Join (Right replaces Left everywhere)

**3)Full Outer Join** Contains results of both Left and Right outer joins.

**Note: Simple join consider as a inner join like**

**Select \* from A join B on (condition);**

**==================================**

**Constraints in DBMS**

Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The whole purpose of constraints is to maintain the **data integrity**during an update/delete/insert into a table. In this tutorial we will learn several types of constraints that can be created in RDBMS.

## Types of constraints

* NOT NULL
* UNIQUE
* DEFAULT
* CHECK
* Key Constraints – PRIMARY KEY, FOREIGN KEY
* Domain constraints
* Mapping constraints

#### NOT NULL:

NOT NULL constraint makes sure that a column does not hold NULL value. When we don’t provide value for a particular column while inserting a record into a table, it takes NULL value by default. By specifying NULL constraint, we can be sure that a particular column(s) cannot have NULL values.

Example:

CREATE TABLE STUDENT(

ROLL\_NO INT **NOT NULL**,

STU\_NAME VARCHAR (35) **NOT NULL**,

STU\_AGE INT **NOT NULL**,

STU\_ADDRESS VARCHAR (235),

PRIMARY KEY (ROLL\_NO)

);

Read more about [this constraint here](https://beginnersbook.com/2014/05/not-null-constraint-in-sql/).

#### UNIQUE:

UNIQUE Constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.

CREATE TABLE STUDENT(

ROLL\_NO INT NOT NULL,

STU\_NAME VARCHAR (35) NOT NULL **UNIQUE**,

STU\_AGE INT NOT NULL,

STU\_ADDRESS VARCHAR (35) **UNIQUE**,

PRIMARY KEY (ROLL\_NO)

);

Read more about it [here](https://beginnersbook.com/2014/05/unique-constraint-in-sql/).

#### DEFAULT:

The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

CREATE TABLE STUDENT(

ROLL\_NO   INT  NOT NULL,

STU\_NAME VARCHAR (35) NOT NULL,

STU\_AGE INT NOT NULL,

EXAM\_FEE INT  **DEFAULT** 10000,

STU\_ADDRESS VARCHAR (35) ,

PRIMARY KEY (ROLL\_NO)

);

Read more: [Default constraint](https://beginnersbook.com/2014/05/default-constraint-in-sql/)

#### CHECK:

This constraint is used for specifying range of values for a particular column of a table. When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

CREATE TABLE STUDENT(

ROLL\_NO   INT  NOT NULL CHECK(ROLL\_NO >1000) ,

STU\_NAME VARCHAR (35)  NOT NULL,

STU\_AGE INT  NOT NULL,

EXAM\_FEE INT DEFAULT 10000,

STU\_ADDRESS VARCHAR (35) ,

PRIMARY KEY (ROLL\_NO)

);

In the above example we have set the check constraint on ROLL\_NO column of STUDENT table. Now, the ROLL\_NO field must have the value greater than 1000.

## Key constraints:

#### PRIMARY KEY:

[Primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/) uniquely identifies each record in a table. It must have unique values and cannot contain nulls. In the below example the ROLL\_NO field is marked as primary key, that means the ROLL\_NO field cannot have duplicate and null values.

CREATE TABLE STUDENT(

ROLL\_NO   INT  NOT NULL,

STU\_NAME VARCHAR (35)  NOT NULL UNIQUE,

STU\_AGE INT NOT NULL,

STU\_ADDRESS VARCHAR (35) UNIQUE,

**PRIMARY KEY** (ROLL\_NO)

);

#### FOREIGN KEY:

Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

#### Domain constraints:

Each table has certain set of columns and each column allows a same type of data, based on its data type. The column does not accept values of any other data type.  
[Domain constraints](https://beginnersbook.com/2015/04/domain-constraints-in-dbms/) are **user defined data type** and we can define them like this:

Domain Constraint = data type + Constraints (NOT NULL / UNIQUE / PRIMARY KEY / FOREIGN KEY / CHECK / DEFAULT)

# Candidate Key in DBMS

 A [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) with no redundant attribute is known as candidate key. Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is that the candidate key should not have any redundant attributes. That’s the reason they are also termed as minimal super key.

## Candidate Key Example

Lets take an example of table “Employee”. This table has three attributes: Emp\_Id, Emp\_Number & Emp\_Name. Here Emp\_Id & Emp\_Number will be having unique values and Emp\_Name can have duplicate values as more than one employees can have same name.

How many super keys the above table can have?  
1. {Emp\_Id}  
2. {Emp\_Number}  
3. {Emp\_Id, Emp\_Number}  
4. {Emp\_Id, Emp\_Name}  
5. {Emp\_Id, Emp\_Number, Emp\_Name}  
6. {Emp\_Number, Emp\_Name}

# Composite key in DBMS

**Definition of Composite key:** A key that has more than one attributes is known as composite key. It is also known as compound key.

**Note:** Any key such as [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/), [primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/), [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) etc. can be called composite key if it has more than one attributes.

# View of Data in DBMS SQL | Views

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition.

We can creating , deleting and updating Views.

Abstraction is one of the main features of database systems. Hiding irrelevant details from user and providing abstract view of data to users, helps in easy and efficient **user-database** interaction. In the previous tutorial, we discussed the [three level of DBMS architecture](https://beginnersbook.com/2018/11/dbms-three-level-architecture/), The top level of that architecture is “view level”. The view level provides the “**view of data**” to the users and hides the irrelevant details such as data relationship, database schema, [constraints](https://beginnersbook.com/2015/04/constraints-in-dbms/), security etc from the user.To fully understand the view of data, you must have a basic knowledge of data abstraction and instance & schema.

* [Data abstraction](https://beginnersbook.com/2015/04/levels-of-abstraction-in-dbms/)
* [Instance and schema](https://beginnersbook.com/2015/04/instance-and-schema-in-dbms/)

**How to see view**

SELECT \* FROM viewName;

**Creating View from a single table:**

Create view view name as Select \* from table;

* In this example we will create a View named DetailsView from the table StudentDetails.  
  Query:

CREATE VIEW DetailsView AS

SELECT NAME, ADDRESS

FROM StudentDetails

WHERE S\_ID < 5;

**Creating View from multiple tables**

CREATE VIEW MarksView AS

SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS

FROM StudentDetails, StudentMarks

WHERE StudentDetails.NAME = StudentMarks.NAME;

**DELETING VIEWS**

DROP VIEW view\_name;

**UPDATING VIEWS(CREATE OR REPLACE VIEW)**

CREATE OR REPLACE VIEW MarksView AS

SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS, StudentMarks.AGE

FROM StudentDetails, StudentMarks

WHERE StudentDetails.NAME = StudentMarks.NAME;

**Inserting a row in a view**

INSERT INTO view\_name(column1, column2 , column3,..)

VALUES(value1, value2, value3..);

Queries:

SELECT \* FROM Employee;

SELECT NAME,salary FROM employee WHERE salary < (SELECT MAX(salary) FROM employee) ORDER BY salary DESC LIMIT 1; -- 2nd highest salary

SELECT MAX(salary) FROM employee GROUP BY department;

SELECT \* FROM employee WHERE salary IN (SELECT MAX(salary) FROM employee GROUP BY department);

SELECT \* FROM employee;

SELECT \* FROM (SELECT MAX(salary) FROM employee) t1 JOIN (SELECT MIN(salary) FROM employee) t2;

SELECT t1.name t1Name,t1.descrip desc1,t2.name t2Name,t2.descrip desc2 FROM test t1,test t2 WHERE (t1.id=t2.ref\_id);

SELECT t1.name t1Name,t1.descrip desc1,t2.name t2Name,t2.descrip desc2 FROM test t1,test t2 WHERE (t1.id=t2.ref\_id);

SELECT \* FROM test;

SELECT department, MAX(salary) FROM employee WHERE salary < (SELECT MAX(salary) FROM employee) GROUP BY department;

SELECT RANK() OVER (ORDER BY salary DESC) sallist FROM employee;

SELECT DISTINCT( salary )

FROM employee Emp1

WHERE N = (SELECT COUNT(DISTINCT ( Emp2.salary ))

FROM employee Emp2

WHERE Emp2.salary >= Emp1.salary);

SELECT DISTINCT( salary )

FROM employee Emp1

WHERE 3 = (SELECT COUNT(DISTINCT ( Emp2.salary ))

FROM employee Emp2

WHERE Emp2.salary >= Emp1.salary);

**=============**