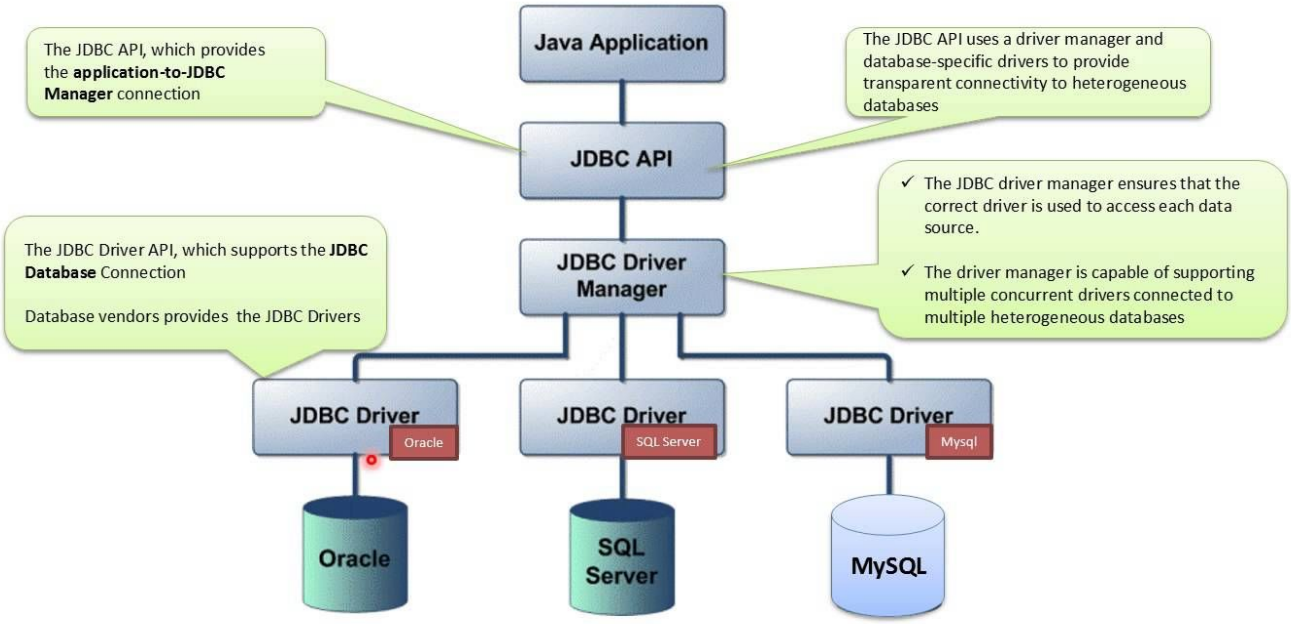
JDBC

Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases.



JDBC Maven Dependency:

```
<dependency>
  <groupId>oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.3</version>
</dependency>
```

3 Important Classes/Steps in JDBC

1. **Connection** helps the java project to connect to the database.
2. **Statement** helps to write and execute SQL queries.
3. **ResultSet** is a Data Structure where we can store the data coming from the database.

Connection String in Detail:

String connection_str =
"jdbc:oracle:thin:@ec2-54-160-198-253.compute-1.amazonaws.com:1521:xe";

jdbc : protocol
oracle : sub-protocol to define database vendors (This might be SQL, MySQL, etc.)
thin : oracle driver name, thin driver
(depends on database itself, for MySQL you don't need this)
host : this is the address of the database server YOUR IP!
Where is your database server? We created our database in the cloud.
port : the door we will get into the database
xe : database service name - uniquely identify our database we're trying to reach

JDBC Set Up:

1. Creating Connection object based on the connection string.

   Connection connection = **DriverManager.*getConnection***(connection_str, db_userName, db_passWord);

2. Once we have a connection object, now we can create a **statement object** from it.

   Statement statement = connection.createStatement();

3. Now, we can use the Statement object to run a query and store the result in a **ResultSet** object.

   ResultSet resultSet = statement.executeQuery(**"SELECT * FROM COUNTRIES"**);

JDBC Methods and Navigation:

Navigation:

- If we want to move backward to access the previous row, We need to set-up the resultSet with "ResultSet.***TYPE_SCROLL_INSENSITIVE*** and ResultSet.***CONCUR_UPDATABLE***".
- They will give us the most updated data and also will help to move backward and forward on the table.
- This is a one time set-up, you just need to know what it is but no detail is required for the interviews.

```
Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                                 ResultSet.CONCUR_UPDATABLE);
```

Methods:

- resultSet.next(); : goes to the NEXT row where the cursor is.
- resultSet.previous(); : goes to the previous row where the cursor stopped.
- resultSet.absolute(2); : moves the cursor to a row that you specified
- resultSet.first(); : goes to the first row regardless of where the cursor is
- resultSet.last(); : goes to the last row regardless of where the cursor is
- resultSet.beforeFirst(); : moves the cursor to the location right before first row
- resultSet.afterLast(); : move the cursor to the location right after last row

1. resultSet.closed(); 2. statement.close(); 3. connection.close();	Connection, statement, and ResultSet are considered as resources. Once you finish working with them, you need to close them. It's like closing your book after you finish reading it. **The order is important!**

What is MetaData?

- MetaData is data about data.
- Metadata Programming is useful to know the capabilities, limitations and facilities of underlying database software and its resources
- DBC metadata programming Supports :
  - DatabaseMetadata
  - ResultSetMetaData

`DatabaseMetaData metaData = connection.getMetaData();`
 	
`System.out.println("User: " + metaData.getUserName());`
`System.out.println("Database Type: "+metaData.getDatabaseProductName());`
`System.out.println("Database Version: "+metaData.getDatabaseProductVersion());`
`System.out.println("Driver Name: "+metaData.getDriverName());`
`System.out.println("Driver Version: "+metaData.getDriverVersion());`
	`User: HR`
	`Database Type: Oracle`
	`Database Version: Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production`
	`Driver Name: Oracle JDBC driver`
	`Driver Version: 12.1.0.1.0`

`ResultSet result = statement.executeQuery("Select * from employees");`
 	
`ResultSetMetaData rsMetadata = result.getMetaData();`
 	
`System.out.println("Columns count: " + rsMetadata.getColumnCount());`
`System.out.println("Column Name: " + rsMetadata.getColumnName(1));`
	`Columns count: 11`
	`Column Name: EMPLOYEE_ID`

SQL Interview Questions:

Do you know SQL?

- Yes, I am very comfortable with writing SQL Queries and DDL and DML commands.
- Currently working with an Oracle database that is running in AMAZON CLOUD SERVER.
- DDL (Data definition language) : CREATE , ALTER, DROP, TRUNCATE
- DML(Data manipulation language): SELECT, DELETE, INSERT, UPDATE

What kind of Database Testing Are You Doing?

- I am mostly doing Database validations.
- I make changes or insert data (create loan) in the front end and validate in the database. Data in the front end matches the DB.
- I also make changes using RESTapi and verify that changes are successful in Database as well.

Do you have any experience with SQL?

- Yes, I have worked with relational databases and I am very comfortable with DDL and DML commands.

Have You Done Any Backend/DataBase testing?

- Yes, I have lots of experience working with databases.
- And I am very comfortable with writing SQL queries.
- I have experience with working on Relational Databases like Oracle, MySQL, SQL Server

Have you worked with non-relational databases?

- I don't have hands on experience, but I know that it is like JSON format
- Database and I have good experience with working with JSON files.
- And I am a quick learner

How can you find 3 HIGHEST PAID employees?

```
SELECT salary, first_name, last_name
FROM employees
ORDER BY salary DESC LIMIT 3;
```

How can you find DUPLICATE NAMES in employees?

```
SELECT first_name, COUNT(first_name)
FROM employees
GROUP BY first_name
HAVING (COUNT(first_name>1));
```

How can you find employees whose salaries are below the average?

```
SELECT first_name, salary
FROM employees
WHERE <= (SELECT AVG (salary) FROM employees);
```

How can you find MAXIMUM salaries in EACH DEPARTMENT?

```
SELECT first_name, MAX(salary)
FROM department d LEFT JOIN employee e ON (d.department_id = e.department_id)
GROUP BY department_id;
```

How can you find the LOWEST Salaries?

```
SELECT first_name, last_name, salary, job_id
FROM employees
WHERE salary =(SELECT MIN (salary) FROM employees);
```

How can you find the SECOND HIGHEST Salary?

```
SELECT MAX (salary)
FROM employees
WHERE salary NOT IN (SELECT MAX(salary) FROM employees);
or
WHERE salary < (SELECT MAX(salary) FROM employees);
```

Get all Employees FIRST_NAME contains 'a' case insensitive

```
SELECT first_name, last_name
FROM employees
WHERE first_name LIKE '%A%' or  first_name LIKE '%a%';
```

- LIKE A% starts with A
- LIKE %A ends with A
- LIKE '%A%' contains A

Count How Many First_Names start with 'A'

```
SELECT COUNT (*) AS CNT // this is to name the column header which will give the result
FROM employees
WHERE first_name LIKE '%A%'
```

Get the count of employees with the salary between 10000 and 20000

```
SELECT COUNT (*)
FROM employees
WHERE salary BETWEEN 10000 AND 20000
```