SQL
# JDBC
Interview
Preparations
Questions

1. **Do you know SQL?**

   - Yes, I am very comfortable with writing SQL Queries and DML commands and exposure to DDL.
   - Currently working with 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

2. **SQL?**

   - Structured Query Language. Used for managing and manipulating data in db.
   - Provide statements for a variety of tasks
     - Querying data
     - Inserting, updating ,deleting rows in a table
     - Creating, replacing, altering, and dropping objects
     - Controlling access to the database and its objects
     - Database consistency and integrity

3. **What are the categories of SQL statements?**
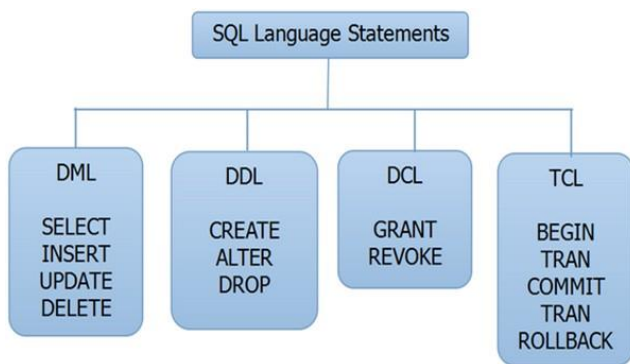
   i. DML (Data Manipulation Language)
   - DML statements affect records in a table. These are basic operations we perform on data such as selecting a few records from a table, inserting new records, deleting unnecessary records, and updating/modifying existing records.

   ii. DDL (Data Definition Language)
   - DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

   iii. DCL (Data Control Language)
   - DCL statements control the level of access that users have on database objects.

   iv. TCL (Transaction Control Language)
   - TCL statements allow you to control and manage transactions to maintain the integrity of data within SQL statements.

4. **Versions**

   - Java 8 → 2014 present Java 7 → 2011 - 2014 Java 6 → 2006 - 2011
     Selenium 3.5.3

5. **Database Schema?**

   - It is like a diagram with all tables and column names, data types and PK, FK and how tables are related to each other

6. **SQL clause?**

   - SELECT and FROM

7. **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 front end matches the DB
   - I also make changes using RESTapi and verify that changes are successful in Database as well.
   - I also support DB migration process. My code connects to Sybase (legacy database) using JDBC then Connects to Oracle (NEW DB) then compare records to make sure data was migrated
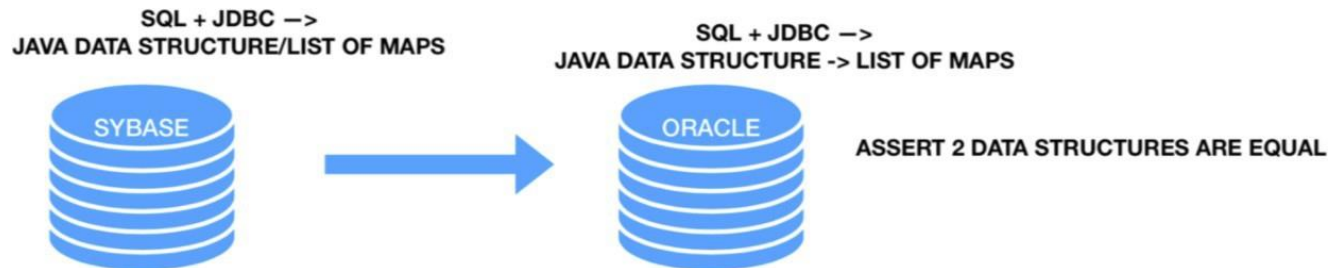
8. **RDBMS**
   - Relational Database Management System
   - Data is organized into tables that are related to each other
     - How are they related?
       - <u>Primary Key</u> (unique and not NULL) and <u>Foreign Key</u> (
     - What type of database system you have expertise with?
       - RDBMS, such as SQL and Oracle

9. **What are constraints?**
   - Properties that table column must comply with.
   - Columns have constraints that defined how data can be stored.
     - <u>Primary Key</u>: unique and NOT NULL
     - <u>Foreign Key</u>: duplicate and NULL and cannot add data which is not in PK
     - <u>Unique Key</u>: only unique value
     - <u>Null</u>: can have null
     - <u>Not null</u>: cannot have null

10. **Data types in SQL?**
    - Number
    - Integers
    - char   → char(20): 20 years spaces are taken from memory
    - varchar → varchar(30): 5 spaces from memory varchar2
    - boolean
    - date
    - currency

11. **Capabilities for SQL select statements**
    - Projection   →Select the columns in a table that are returned by a query
    - Selection   →Selects the rows in a table that are returned by a query
    - Join   →Brings together data that is stored in different tables by specifying the link between them

12. DML (Data Manipulation Language) vs DDL (Data Definition Language)

DML command actions can be restored.

Commands:
- **SELECT** from tablename; (read)
- **INSERT** into tablename values (...); (add)
- **UPDATE** tablename SET value WHERE location;
- **DELETE** from tablename WHERE location; (rows)
- **MERGE**

13. SQL JOIN JOINS

JOIN (INNER) JOIN

is used when retrieving data from multiple tables and will return only matching data
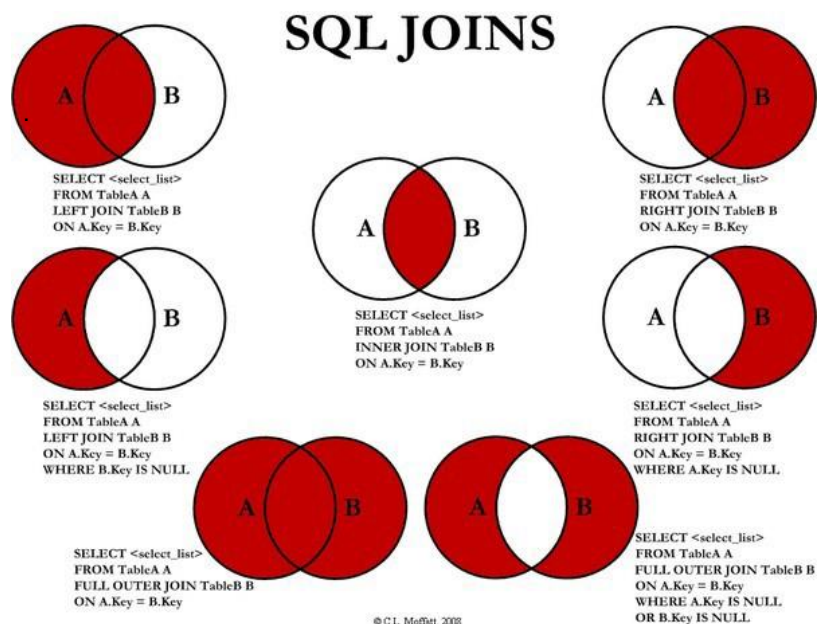
LEFT (OUTER) JOIN

is used when retrieving data from multiple tables and will return left table and any matching right table records.

RIGHT (OUTER) JOIN

is used when retrieving data from multiple tables and will return right table and any matching left table records.

FULL (OUTER) JOIN

is used when retrieving data from multiple tables and will return both table records, matching and non-matching

SQL JOINS

SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL

SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL

SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL

© C.L. Moffatt, 2008

14. UNION
- Union combines the resultSets of two queries

```
select from column_names from table_name {UNION} select column_name from table_name)
```

15. Find duplicate names in employees

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

16. How to find employees whose salaries are below the average?

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

17. How to find maximum salaries in each department?

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

18. How to find lowest salaries?

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

19. How to find second highest salary of employees?

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

20. SQL Developer
- Development environment (manual testing the database using the queries)
  - Release 2.1 -2009 ☐ 3.0 - 2011
  - Release 4.0 - 2013(latest)
- Has a Migration release(1.2) → provides users with a single point to browse data in third-party DB and to migrate from these DB to Oracle
- Supports Window, Linux and Mac OS x

21. Writing SQL Statements
- Keywords are uppercased while columns and table names are lowercase
- Statements are not case sensitive
- Clauses are usually placed on separate lines
- Keywords cannot be abbreviated or split across lines

22. Arithmetic Expressions
- You use the operators in any clause (except the From clause)
- With Date and Timestamp - can only use addition and subtraction
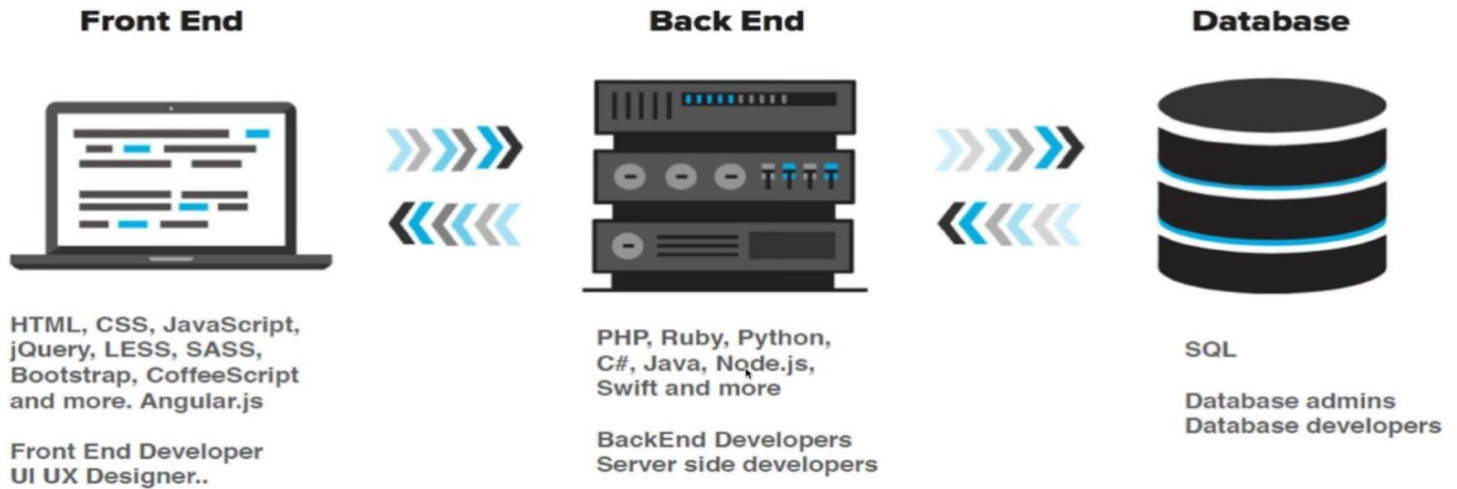- Add (+), Subtract (-), Multiply (*), Divide (/)

23. Do you have experience with SQL?
- Yes, I have worked with relational databases and i am very comfortable with and DML commands

24. Back-end test framework
- Using RESTASSURED in my framework in order for the process of Deserialization and serialization to occur that way you store Json response into a java collection data structure and assert the data with the expected value (also stored in java data structure)
- I use postman for manual testing first BEFORE I INVOKE MY FRAMEWORK

25. Back-end Testing

Front End

HTML, CSS, JavaScript, jQuery, LESS, SASS, Bootstrap, CoffeeScript and more. Angular.js

Front End Developer
UI UX Designer..

Back End

PHP, Ruby, Python, C#, Java, Node.js, Swift and more

BackEnd Developers
Server side developers

Database

SQL

Database admins
Database developers

26. Have you done any backend/database testing?
- Yes, I have lots of experience with 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

27. 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

28. Database testing framework
- For manual testing I use SQL developer for producing SQL queries
- FOR AUTOMATION; I use JDBC library to integrate java by getting a CONNECTION from oracle database then creating STATEMENTS using SQL queries and then storing the data into a RESULTSET object.

29. What is metadata?
- MetaData Data about Data

```
ResultSetMetaData rsmd = rs.getMetaData();
int columNum=rsmd.getColumnCount();
```

30. Data Structures and Why we need it?
- Data structures are way of organizing data for efficient
- manipulation: Insertion , searching, reading , deletion of data.
- I always use java data structures for reading data and storing data from our application, database, or API.

31. How can we connect DB?

```
Connection connection = DriverManager.getConnection(URL, user,password);
Statement statement = connection.createStatement();
ResultSet resultset = statement.executeQuery("query");

resultset.close();
statement.close();
connection.close();

// After connection;
DatabaseMetaData db = connection.getMetaData();

// After resultSet;
ResultSetMetaData rs = resultSet.getMetaData();
```

- If failure to connect will throw an exception:
  - SQLException   (bad URL or credentials)
  - ClassNotFoundException (JDB driver not in classpath)

32. Dependency for JDBC

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

33. What is SQL Constraints?
- SQL Constraints are rules used to limit the type of data that can go into a table, to maintain the accuracy and integrity of the data inside table.
- Constraints can be divided into the following two types,
  - **Column level constraints**: Limits only column data.
  - **Table level constraints**: Limits whole table data.
- Constraints are used to make sure that the integrity of data is maintained in the database. Following are the most used constraints that can be applied to a table.
  - ☐ **NOT NULL** ☐ constraint restricts a column from having a **NULL** value

    ```
    CREATE TABLE Student(s_id int NOT NULL, Name varchar(60), Age int);
    ```

  - ☐ **UNIQUE** ☐ constraint ensures that a field or column will only have **unique values.**
    - Constraint when creating a table (Table level)

      ```
      CREATE TABLE Student(s_id int NOT NULL UNIQUE, Name varchar(60), Age int);
      ```
    - Constraint after creating a table (Column level)

      ```
      ALTER TABLE Student ADD UNIQUE(s_id);
      ```

  - ☐ **PRIMARY KEY** ☐ constraint uniquely identifies each record in a database. Unique and not null
    - Constraint at Table level

      ```
      CREATE TABLE Student(s_id int PRIMARY KEY, Name varchar(60) NOT NULL, Age int);
      ```
    - Constraint at Column level

      ```
      ALTER TABLE Student ADD PRIMARY KEY (s_id);
      ```

  - ☐ **FOREIGN KEY** ☐ is used to relate two tables. **FOREIGN KEY** constraint is also used to restrict actions that would destroy links between tables. Let's see its use, with help of the below tables:

Customer_Detail Table

c_id	Customer_Name	address
101	Adam	Noida
102	Alex	Delhi
103	Stuart	Rohtak

Order_Detail Table

Order_id	Order_Name	c_id
10	Order1	101
11	Order2	103
12	Order3	102

*In Customer_Detail table, **c_id** is the primary key which is set as foreign key in Order_Detail table.*
If you try to insert any incorrect data, DBMS will return error and will not allow you to insert the data.

o   Using FOREIGN KEY constraint at Table Level

```
CREATE table Order_Detail(
    order_id int PRIMARY KEY,
    order_name varchar(60) NOT NULL,
    c_id int FOREIGN KEY REFERENCES Customer_Detail(c_id)
);
```

In this query, c_id in table Order_Detail is made as foreign key, which is a reference of c_id column in Customer_Detail table.

o   Using FOREIGN KEY constraint at Column Level

```
ALTER table Order_Detail ADD FOREIGN KEY (c_id) REFERENCES Customer_Detail(c_id);
```

☐ **CHECK**: constraint is used to restrict the value of a column between a range. It performs check on the values, before storing them into the database. It's like condition checking before saving data into a column.

o   Using **CHECK** constraint at Table Level

```
CREATE table Student(
    s_id int NOT NULL CHECK(s_id > 0),
    Name varchar(60) NOT NULL,
    Age int
);
```

The above query will restrict the s_id value to be greater than zero.

o   Using **CHECK** constraint at Column Level

```
ALTER table Student ADD CHECK(s_id > 0);
```

☐ **DEFAULT**