CLOB:

CLOB operations can be very useful for storing and manipulating large amounts of text data. They can be used to store documents, articles, blog posts, and other types of text data.

When inserting or updating CLOB data, be sure to use the appropriate parameter type. For example, if you are using a JDBC driver, you should use the setAsciiStream() or setClob() method to set CLOB parameters.When selecting CLOB data, be aware that some database systems may limit the amount of CLOB data that can be returned in a single row. If you need to select a large amount of CLOB data, you may need to use a streaming cursor or a similar technique.

When performing complex operations on CLOB data, be sure to test your queries thoroughly to make sure that they are working as expected.

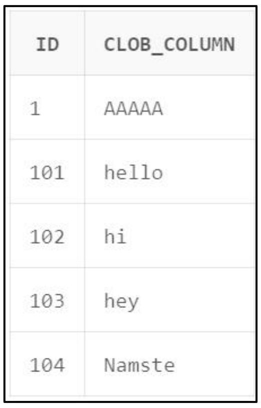
SQL Query:

```
CREATE TABLE MyTable (
        id INT PRIMARY KEY,
   clob_column CLOB NOT NULL
);
INSERT INTO MyTable(id, clob_column) VALUES (1, EMPTY_CLOB());
INSERT INTO MyTable(id, clob_column) VALUES (101,to_clob('hello'));
INSERT INTO MyTable(id, clob_column) VALUES (102,to_clob('hi'));
INSERT INTO MyTable(id, clob_column) VALUES (103,to_clob('hey'));
INSERT INTO MyTable(id, clob_column) VALUES (104,to_clob('Namste'));

UPDATE MyTable
SET clob_column = 'AAAAA'
WHERE id = 1;
```

-- Select a data from a table.

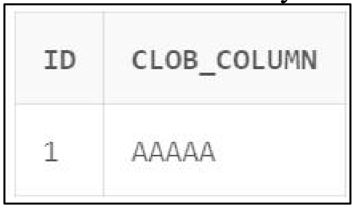
```
SELECT * FROM MyTable;
```



ID	CLOB_COLUMN
1	AAAAA
101	hello
102	hi
103	hey
104	Namste

-- Select a CLOB from a table.

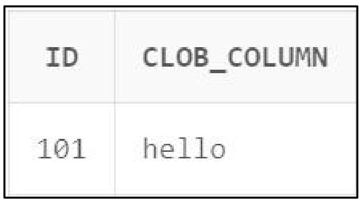
```
SELECT * FROM MyTable WHERE id = 1;
```



ID	CLOB_COLUMN
1	AAAAA

-- Search for a pattern in a CLOB.

```
SELECT * FROM MyTable WHERE clob_column LIKE '%hello%';
```



ID	CLOB_COLUMN
101	hello

BLOB:

BLOB operations can be very useful for storing and manipulating large amounts of data. They can be used to store images, audio files, videos, and other types of data. By using BLOB operations in SQL queries, you can efficiently and reliably store and manage large amounts of data.

SQL Query:

```
CREATE TABLE MyTable (
   id INT PRIMARY KEY,
   blob_column BLOB NOT NULL
);

INSERT INTO MyTable(id, blob_column) VALUES (1, EMPTY_BLOB());

UPDATE MyTable
  SET blob_column = 'VVVVV'
WHERE id = 1;
```

BFILE:

BFILEs are useful for storing large amounts of data that is not frequently accessed, such as images, videos, and audio files. They can also be used to store data that is too large to fit in a database column, such as a large text file or a binary file.

The BFILE data type in SQL is a large binary object that is stored in an external file outside of the database. BFILEs are read-only and cannot be updated or deleted directly from within the database.

LOAD FILE: This operation loads the contents of a file into a database table.

SAVE FILE: This operation saves the contents of a database table to a file.

APPEND FILE: This operation appends the contents of a database table to the end of a file.

COPY FILE: This operation copies the contents of a file to another file.

DELETE FILE: This operation deletes a file.

SQL Query:

Creating Tables Containing BFILE Objects

```
 CREATE TABLE myTable (
     id           INTEGER PRIMARY KEY,
     bfile_column BFILE NOT NULL
   );
```

Populating BFILE column

```
INSERT INTO myBFile VALUES (1,BFILENAME('BFILE_DIR','test.bmp'));
```