INDEXING IN SQL

Indexing in databases is a technique used to improve the performance of queries by allowing for efficient retrieval of data. It involves creating data structures that store a subset of the data from a table, organized in a way that makes it quicker to search, filter, and sort the data based on specific criteria.

BEFORE INDEXING

```
mysql> select * from dd where location='Bhuj';
```

```
| RESTRAUNT      | Bhuj      |  4
+----------------+-----------+---
22027 rows in set (0.56 sec)
```

AFTER INDEXING

```
mysql> create index arp on dd(Location);
Query OK, 0 rows affected (2.42 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> select * from dd where location='Bhuj';
```

```
+----------------+-----------+---
22027 rows in set (0.19 sec)
```

```
mysql> show index from dd;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-------
-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expres
sion |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-------
-----+
| dd    |          1 | arp      |            1 | Location    | A         |          47 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL
     |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+-------
-----+
1 row in set (0.00 sec)
```

There are several other types of indexing commonly used in databases. Here are some of them with examples:

1)**Unique indexing:** Unique indexing in databases is a feature that ensures the uniqueness of values in one or more columns of a table. It prevents duplicate entries from being inserted into the indexed column(s). This can be particularly useful for columns that should contain unique values, such as primary keys or email addresses

```
mysql> CREATE UNIQUE INDEX uniindx ON dd(cc(255)); -- Specify a key length of 255 characters
Query OK, 0 rows affected (1.10 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

2) **Composite Index**: A composite index involves multiple columns and is useful for queries that filter or sort data based on multiple criteria.

```
mysql> CREATE INDEX AA ON DD(Domain,Value);
Query OK, 0 rows affected (2.97 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3) **Clustered Index**: A clustered index is a type of index where the rows of the table are physically stored in the same order as the index. In MySQL's InnoDB storage engine, the primary key is implicitly clustered.

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    INDEX idx_customer_id (customer_id)
) ENGINE=InnoDB;
```

>DROP INDEX

```
mysql> DROP INDEX arp ON dd;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```