Dropping Indexes in SQL

What are Indexes?

- Indexes in SQL are database objects that are used to improve the speed of data retrieval operations on a table
- They are created on one or more columns of a table, and they allow the database engine to find rows more quickly and efficiently
- Types of Indexes : Primary Key, Unique Index, Clustered Index, Non-Clustered Index, Covering Index, Full-Text Index, Filtered Index, Spatial Index, XML Index, Hash Index, and Bitmap Index

Indexes in MySQL?

Example : Employees Table

```
1 •  select * from employees;
2
```

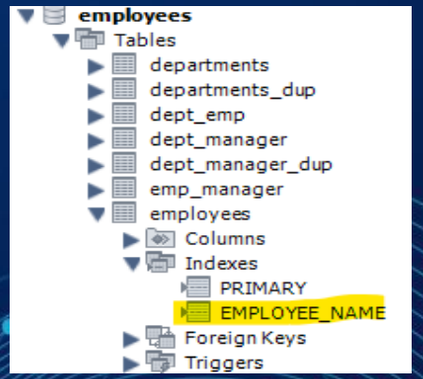
Result Grid | Filter Rows: | Edit: | Export/Import

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24

Syntax to create index

```
CREATE INDEX EMPLOYEE_NAME
ON EMPLOYEES (FIRST_NAME, LAST_NAME);
```

Place to find created Index



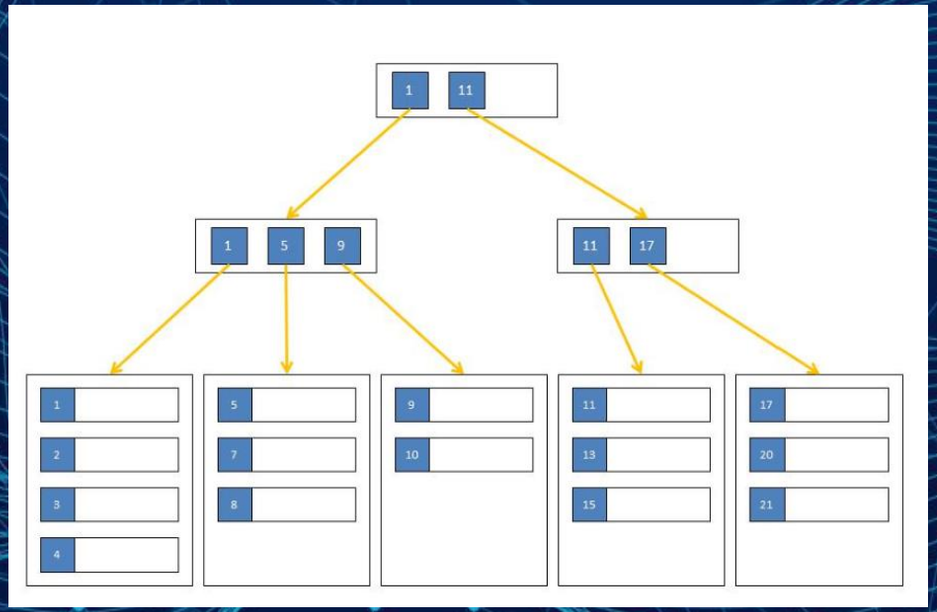

```
3 •  show index from employees;
4
5
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
employees	0	PRIMARY	1	emp_no	A	298980	NULL	NULL		BTREE			YES

How Indexes work?

- An index creates an entry for each value that appears in the indexed columns.
- These entries are stored in a data structure (usually a B-tree or hash table) that allows for fast searches, lookups, and sorted order.



Benefits of Indexes

- Faster query performance, especially on large tables.
- Efficient retrieval of rows that match certain criteria.

Drawbacks of Indexes

- Additional storage space required for the index data.
- Slower performance for INSERT, UPDATE, and DELETE operations due to the overhead of maintaining the index.

Why Drop Indexes?

- Improve database performance
- Remove redundant or unused indexes
- Adapt to changing data requirements
- Optimize storage space

Syntax to Drop Indexes in SQL

- Drop Index Index_name on table_name;

```
DROP INDEX EMPLOYEE_NAME
ON EMPLOYEES;
```

Pre-Drop Considerations

- Assess the impact on query performance.
- Ensure no dependencies on the index
- Backup the current state of the database
- Use tools to analyze index usage (e.g., DMVs in SQL Server)


Thank you!