



MySQL - RDBMS

Trainer: Mr. Rohan Paramane



Index

- Index enable faster searching in tables by indexed columns.
 - `CREATE INDEX idx_name ON table(column);`
- One table can have multiple indexes on different columns/order.
- Typically indexes are stored as some data structure (like BTREE or HASH) on disk.
- Indexes are updated during DML operations. So DML operation are slower on indexed tables.



Index

- Index can be ASC or DESC.
 - It cause storage of key values in respective order (MySQL 8.x onwards).
 - ASC/DESC index is used by optimizer on ORDER BY queries.
- There are four types of indexes:
 - Simple index
 - `CREATE INDEX idx_name ON table(column [ASC|DESC]);`
 - Unique index
 - `CREATE UNIQUE INDEX idx_name ON table(column [ASC|DESC]);`
 - Doesn't allow duplicate values.
 - Composite index
 - `CREATE INDEX idx_name ON table(column1 [ASC|DESC], column2 [ASC|DESC]);`
 - Composite index can also be unique. Do not allow duplicate combination of columns.
 - Clustered index
 - PRIMARY index automatically created on Primary key for row lookup.
 - If primary key is not available, hidden index is created on synthetic column.
 - It is maintained in tabular form and its reference is used in other indexes.



Index

- Indexes should be created on shorter (INT, CHAR, ...) columns to save disk space.
- Few RDBMS do not allow indexes on external columns i.e. TEXT, BLOB.
- MySQL support indexing on TEXT/BLOB up to n characters.
 - CREATE TABLE test (blob_col BLOB, ..., INDEX(blob_col(10)));
- To list all indexes on table:
 - SHOW INDEXES FROM table;
- To drop an index:
 - DROP INDEX idx_name ON table;
- When table is dropped, all indexes are automatically dropped.
- Indexes should not be created on the columns not used frequent search, ordering or grouping operations.
- Columns in join operation should be indexed for better performance.



Query performance

- Few RDBMS features ensure better query performance.
 - Index speed up execution of SELECT queries (search operations).
 - Correlated sub-queries execute faster.
- Query performance can be observed using EXPLAIN statement.
 - EXPLAIN FORMAT=JSON SELECT ...;
- EXPLAIN statement shows
 - Query cost (Lower is the cost, faster is the query execution).
 - Execution plan (Algorithm used to execute query e.g. loop, semi-join, materialization, etc).
- Optimizations can be enabled or disabled by optimizer_switch system variable.
 - SELECT @@optimizer_switch;
 - SET @@optimizer_switch='materialization=off';



Constraints

- Constraints are restrictions imposed on columns.
- There are five constraints
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK
- Few constraints can be applied at either column level or table level. Few constraints can be applied on both.
- Optionally constraint names can be mentioned while creating the constraint. If not given, it is auto-generated.
- Each DML operation check the constraints before manipulating the values. If any constraint is violated, error is raised.



Constraints

- NOT NULL

- NULL values are not allowed.
- Can be applied at column level only.
- CREATE TABLE table(c1 TYPE NOT NULL, ...);

- UNIQUE

- Duplicate values are not allowed.
- NULL values are allowed.
- Not applicable for TEXT and BLOB.
- UNIQUE can be applied on one or more columns.
- Internally creates unique index on the column (fast searching).
- A table can have one or more unique keys.
- Can be applied at column level or table level.
 - CREATE TABLE table(c1 TYPE UNIQUE, ...);
 - CREATE TABLE table(c1 TYPE, ..., UNIQUE(c1));
 - CREATE TABLE table(c1 TYPE, ..., CONSTRAINT constraint_name UNIQUE(c1));



Constraints

- PRIMARY KEY

- Column or set of columns that uniquely identifies a row.
- Only one primary key is allowed for a table.
- Primary key column cannot have duplicate or NULL values.
- Internally index is created on PK column.
- TEXT/BLOB cannot be primary key.
- If no obvious choice available for PK, composite or surrogate PK can be created.
- Creating PK for a table is a good practice.
- PK can be created at table level or column level.
- `CREATE TABLE table(c1 TYPE PRIMARY KEY, ...);`
- `CREATE TABLE table(c1 TYPE, ..., PRIMARY KEY(c1));`
- `CREATE TABLE table(c1 TYPE, ..., CONSTRAINT constraint_name PRIMARY KEY(c1));`
- `CREATE TABLE table(c1 TYPE, c2 TYPE, ..., PRIMARY KEY(c1, c2));`





Thank you!

Rohan Paramane<rohan.paramane@sunbeaminfo.com>

