SB101

Day-12

INDEX & NORMALIZATION

INDEX

An index is a data structure such as B-Tree that improves the speed of data retrieval on a table at the cost of additional writes and storage to maintain it. The query optimizer may use indexes to quickly locate data without having to scan every row in a table for a given query.

❑ Primary Key and Unique key are types of index.
❑ Index can be applied on a single column or combination of more than column.
❑ By default, MySQL creates the B-Tree index if you don’t specify the index type.

```
CREATE TABLE t(
  col-name data-type(size) constraint,
  col-name data-type(size) constraint,
  …
  col-name data-type(size) constraint,
  INDEX (col-name-1, col-name-1)
);
```

INDEX

To add an index for a column or a set of columns, you use the CREATE INDEX statement as follows:

```
CREATE INDEX index-name ON table-name (column-list);
```

To view list of existing index in the table-

```
SHOW INDEX FROM table-name;
```

To drop index-

```
ALTER TABLE table-name DROP INDEX index-name;
```

Normalization

Insertion Anomaly: Add fake/repeated entries to maintain some information and it leads to data inconsistency.

Updation Anomaly: To update a small information, one has to update multiple records.

Deletion Anomaly: Deletion of one information leads to deletion of another type of information.

A well structured relations is a relation, which contains minimum data redundancy and allow users to insert, delete and update rows without causing data inconsistencies. To create such efficient tables, normalization is used.

Caution: Using normalization it is not guaranteed that redundancy & anomalies will be eliminated completely but they can be minimized so summarization is; When tables are normalized then-

❑ Complexity & Access time: High

❑ Redundancy & Anomalies: Low

candidate key v/s primary key

Primary key	Candidate key
Primary Key is a unique and non-null key which identify a record uniquely in table. A table can have only one primary key.	Candidate key is also a unique key to identify a record uniquely in a table but a table can have multiple candidate keys.
Primary key column value can not be null.	Candidate key column can have null value.
Primary key is most important part of any relation or table.	Candidate key signifies as which key can be used as Primary Key.
Primary Key is a candidate key.	Candidate key may or may not be a primary key.

The columns in a candidate key are called **prime attributes**, and a column that does not occur in any candidate key is called a **non-prime attribute**.

Normalization

We are having following normal forms.
(A) 1-NF (B) 2-NF (C) 3-NF

❑ As the level of normalization increases then the number of tables also increase and the same is for complexity.

❑ To achieve the nth normal form the table must be in the (n-1)th normal form so we can say that if table is in the nth normal form then it will be in the (n - 1)th normal form or we can say that nth normal form is superset of (n - 1)th normal form.

1-NF

❑ No multi-values attributes are allowed

❑ Each attribute value must be atomic (Not further decomposible)

Functional Dependency

In a relation R, we have two attributes that are A and B, the attribute B is said to be functionally dependent on attribute A if B can be uniquely identified using value of A or we can say that each value of A is associated with exactly one value of B. Functional dependency between A and B is represented as A -> B. A is called determinant and B is called dependent.

For example:
Student name is functionally dependent on roll no [roll no -> Student name]
tax payer details are functionally dependent on PAN [PAN -> tax payer details]

Fully Functional Dependency

If X and Y are attribute set in a relation, Y is fully functional dependent on X, if Y is functionally dependent on X not on any proper subset of X

Example ABC -> D (D is fully functionally dependent on combination of A,B and C not on any subset of A,B or C)
A -> D [False] B -> D [False] C -> D [False]
AB -> D [False] BC -> D [False] CA -> D [False]

Functional Dependency & 2-NF

Partial Functional Dependency

If X and Y are attribute set in a relation, Y is partially functional dependent on X, if Y is functionally dependent on any proper subset of X

Example ABC -> D (For partial dependency, any of the following should be true)
A -> D, B -> D, C -> D, AB -> D, BC -> D, CA -> D

2-NF

- ❑ The table must be in 1-NF
- ❑ No partial dependency will be there i.e. Non prime attributes should not be dependent on the subset of a candidate key.
- ❑ We have to decompose the table to remove partial dependency

Transitive Dependency & 3-NF

Transitive Functional Dependency

In transitive functional dependency, dependent is indirectly dependent on the determinant i.e. a -> b and b -> c the according to the axiom of transitivity a -> c this is called transitive functional dependency.

3-NF

- ❑ The table must be in 2-NF
- ❑ Must not have transitive dependency
- ❑ We have to further decompose the table to remove transitive dependency

QUERY?