→ Indexes:

a. Indexes are used to retrieve data from database more quickly
b. The user cannot see the indexes, this is only used to speed up the process of searching and sorting
c. An index uses data structures such as B-Tree, which improves the speed of data retrieval on a table at the cost of additional rights and storage to maintain
d. Indexes are quick to locate the data, without having to scan every row in a table for a given query
e. When you create a table with a primary key or unique key, MySQL automatically creates a special index name primary. This index can be used/called clustered index
f. The primary index is special because the index itself is stored together <complete it>
g. The clustered index enforces the order of rows in the table
h. Indexes other than primary index are called secondary index or non-clustered index
i. While creating index, one must choose the appropriate column to improve the performance of retrieval
j. It is important to use the index wisely as it may lead to impact the performance of write operation of inserting, updating and deleting

→ Advantages of indexing

a. Index helps us to avoid duplicate row data
b. Text searching - you can search any text or phrases within a large amount of text data
c. It optimizes your query

→ Types of indices:

a. There are different types of indexes:
    i. B-Tree index:
        1. This is the most commonly used index
        2. It organizes the data in tree-like structures for efficient searching
        3. This allows faster retrieval of data based on single or multiple columns
        4. This can be used for both , exact match and range query
        5. Syntax for B-Tree index:

```
CREATE INDEX index_name ON table_name(column1_name);
CREATE INDEX index_name ON table_name(column1_name,
column2_name);
```

    ii. Hashed index:
        1. This index is used to search exact match data
        2. These are used to search exact match on columns with fixed length, such as integer
        3. To store data, this index uses a hash–table data structure, which allows fastest retrieval of data based on a single column
        4. This below example creates hash index on emp_id column
        5. Example for Hashed index:

```
CREATE TABLE employees(
emp_id INT PRIMARY_KEY,
emp_name VARCHAR(50),
INDEX index_name(emp_id) USING HASH
);
```

iii. Full text index:

1. This index is used for searching through text data in a table
2. This allows fastest retrieval of data based on keywords or phrases search
3. This below example will create index on book_name and book_content
4. Syntax for Full text index:

```
CREATE TABLE ebook(
book_id INT PRIMARY_KEY,
book_name VARCHAR(50),
book_content text,
FULLTEXT index_name(book_name, book_content) USING HASH
);
```

iv. Spatial index:

1. This index is used to search through data that has spatial component like latitude, longitude
2. It allows fast retrieval of data based on spatial data such as finding all points within a certain distance of specific location
3. Syntax for Spatial index:

```
CREATE TABLE location(
location_id INT PRIMARY_KEY,
location_p POINT,
SPATIAL INDEX index_name(location_p)
);
```

v. Clustered index:

1. Clustered index is used to physically order the data in a table based on the index key
2. This can improve the performance of a query that can access a large portion of data in a table as data is stored in the same order as the index key
3.

vi. sasa

b. asa

→

→ imp

a.
b. Stored procedures
c. Functions
d. Cursor
e. Triggers
f. routines