



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

# AUTOMATIC INDEX CREATION

**Saksham Rathi, Kavya Gupta, Shravan S, Mayank Kumar**  
(22B1003) (22B1053) (22B1054) (22B0933)

CS349: DATABASE AND INFORMATION SYSTEMS  
UNDER PROF. SUDARSHAN AND PROF. SURAJ

Indian Institute of Technology Bombay  
Spring 2024-25

# Contents



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

- 1 Introduction
- 2 Directory Structure
- 3 When to use indices?

# Introduction to the Problem Statement



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

- Indexes are crucial for efficient query execution in relational databases.
- However, developers sometimes forget to create indexes for frequently queried columns.
- This can lead to repeated full relation scans, significantly degrading performance.
- **Goal:** Modify the application layer of PostgreSQL to detect such patterns and automatically create indexes when beneficial.
- Approach:
  - Track full relation scans with equality predicates.
  - Estimate the potential benefit of an index.
  - Automatically trigger index creation if estimated benefit outweighs the cost.
  - Rejecting low selectivity columns, such as gender, which has low number of distinct values.

# Directory Structure



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

Here is the directory structure of the submission:

- `./code`: Contains the header and C++ files for the implementation, along with the Makefile.
- `./theory`: Contains some relevant paper and slides.
- `./documentation`: Contains the report as `readme.pdf`.
- `./README.md`: Contains the instructions to run the code.

# About Indices



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

An index in SQL is a database object that improves the speed of data retrieval operations on a database table.

When a query is executed, the database can use the index to quickly find the relevant rows.

Without an index, the database might need to scan every row to find the data, which is much slower.

# When to use indices?



DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?

- **Frequent searches on specific columns:** Columns that are often used in WHERE clauses, JOIN conditions or as part of a SELECT query.
- **Large Tables with Heavy Read Operations:** Tables with a vast number of records where read operations are more common than write operations.
- **Columns used in JOINS:** Indexing these columns can speed up the join process.
- **Unique or Primary Key Constraints:** Indices improve lookup efficiency, so easy to impose such constraints.
- **Composite Indices:** When queries often filter on multiple columns, a composite index can be beneficial, rather than creating separate indices for each column.



# When to use indices?

There are also cases, where we should refrain from using indices, such as tables with heavy write operations, because indices slow down INSERT, UPDATE, and DELETE operations (index needs to be updated too). Similarly, in case of small tables, or columns with low selectivity (many duplicate values).

Indices, overall lead to improved query performance, slower write operations, and increased storage requirements.

We can analyze how a query is executed, and whether an index is effectively used or not by using the EXPLAIN command in PostgreSQL. Moreover, to maintain performance, especially in databases with frequent data modifications, we need to regularly rebuild and reorganize indices.

DBIS

[Authors](#)

Introduction

Directory  
Structure

When to use  
indices?