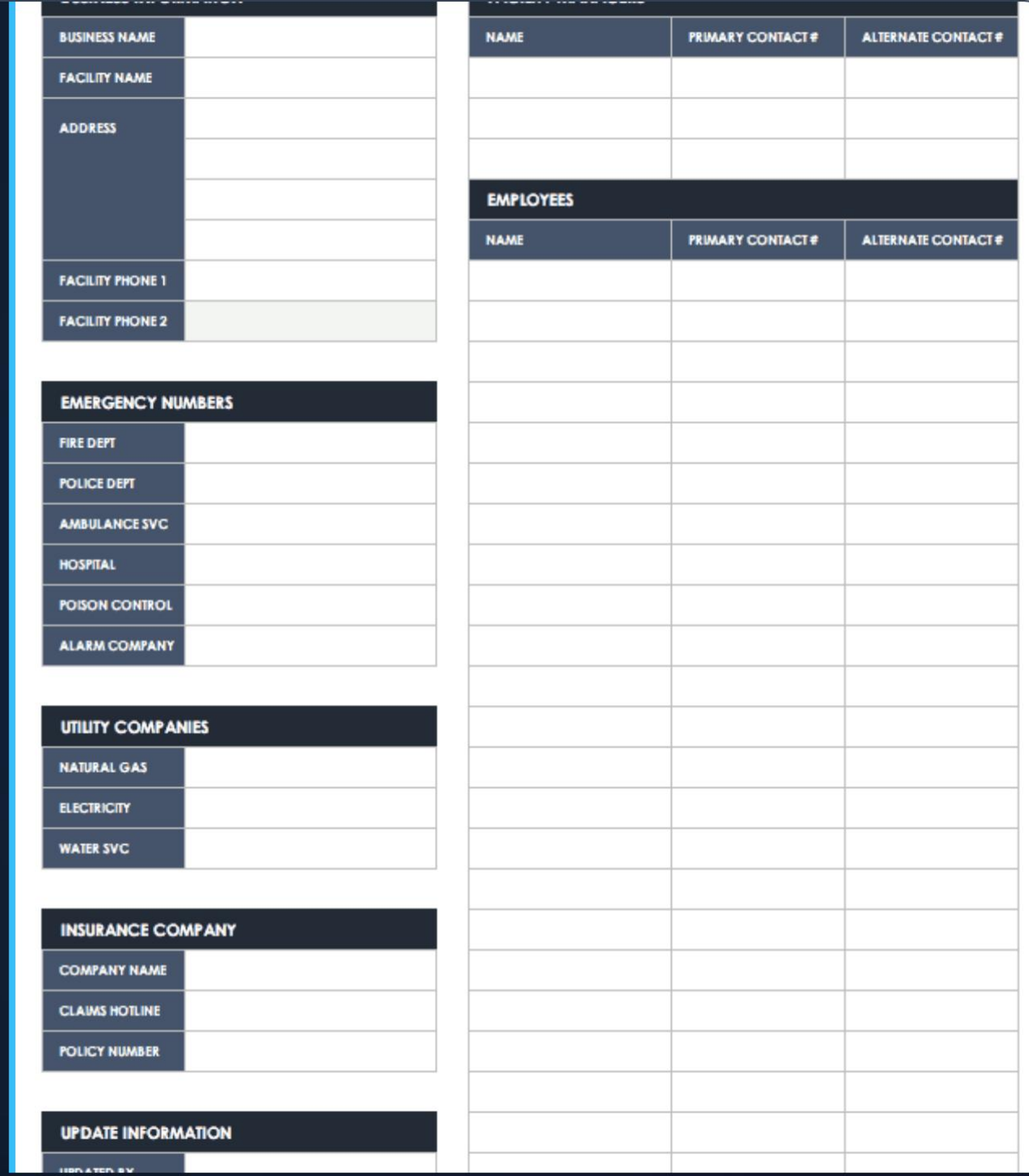
POSTGRESQL BASICS

Apa Itu Index di Database?

Bayangkan sebuah **Buku Telepon** dengan jutaan nama.

Tanpa index, Anda harus mencari satu per satu dari halaman pertama (*Full Table Scan*).

Dengan index, Anda langsung melompat ke huruf yang dicari.



Cara Kerja & Mitos

Index ≠ Caching

Ini adalah perbedaan fundamental:

- **Index:** Struktur data permanen di disk (Daftar Isi). Mempercepat *pencarian*.
- **Cache:** Salinan data sementara di RAM. Mempercepat *akses ulang*.

Di Dalam Postgres

Saat Anda menjalankan:

```
CREATE INDEX ON users(name);
```

Postgres membangun struktur (B-Tree) yang menyimpan:

1. Nilai kolom (misal: "Budi")
2. Pointer lokasi data (Tuple ID)

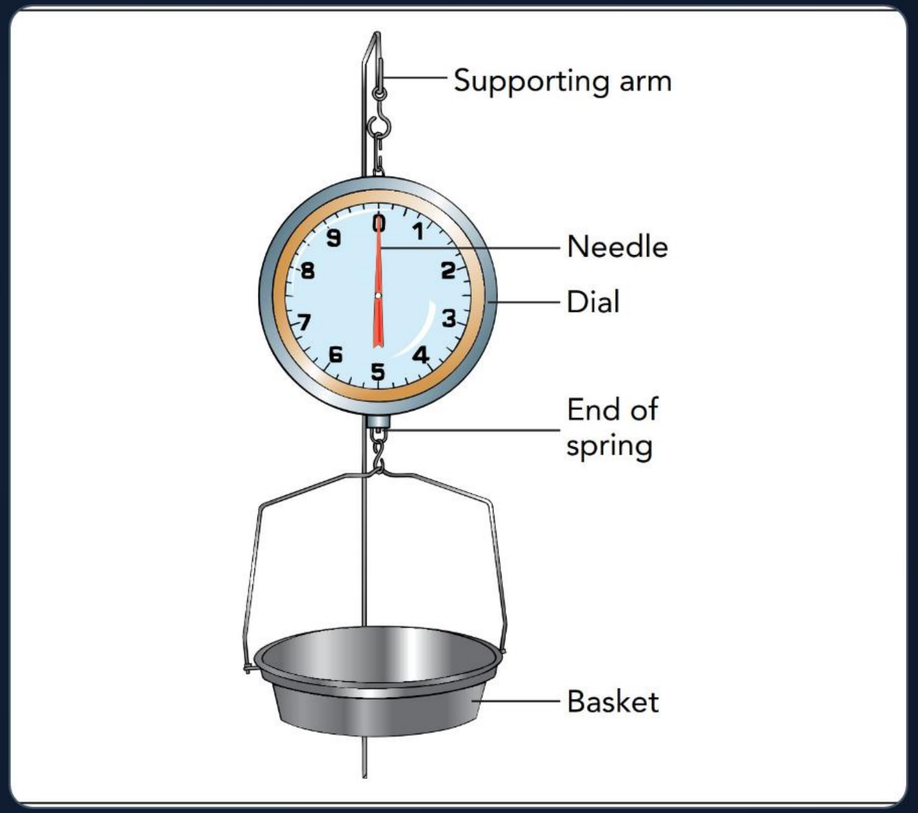
Hasil: Lompat langsung ke baris yang tepat.

Mengapa Index Tidak Gratis?

Index adalah sebuah **Trade-off** (Pertukaran).

- ✕ **Memakan Ruang:** Setiap index butuh penyimpanan disk tambahan.
- ✕ **Memperlambat Penulisan:** Setiap kali ada `INSERT`, `UPDATE`, atau `DELETE`, database harus memperbarui tabel UTAMA dan semua INDEX-nya.

> **Analogi:** Anda punya 5 daftar isi berbeda untuk satu buku. Saat isi buku berubah, Anda harus merevisi kelima daftar isi tersebut.



Jenis Index Dasar

1. B-TREE (Default)

99% Kebutuhan Umum

- Cocok untuk pencarian persis (`=`)
- Cocok untuk rentang (`<` , `>` , `BETWEEN`)
- Mendukung sorting (`ORDER BY`)

2. HASH Index

Spesifik & Cepat

- Hanya untuk pencarian sama dengan (`=`)
- Sangat cepat untuk operasi equality.
- **Kekurangan:** Tidak bisa untuk range atau sorting. Jarang dipakai dibanding B-Tree.

Index Untuk Data Kompleks

GIN

Generalized Inverted Index

Seperti indeks kata di belakang buku. Cocok untuk **JSONB**, **Array**, dan **Full Text Search**.

GiST

Generalized Search Tree

Struktur pohon fleksibel. Ideal untuk data **Geospasial (PostGIS)** dan pencarian "tetangga terdekat".

BRIN

Block Range Index

Untuk **Big Data** berurutan (misal: Log Timestamp). Sangat kecil, hanya menyimpan ringkasan per blok.

Teknik Optimasi Lanjutan

Partial Index

Index hanya untuk subset data tertentu.

```
WHERE is_active = true
```

Manfaat: Ukuran index jauh lebih kecil dan update lebih ringan. Cocok jika Anda sering query hanya sebagian data.

Expression Index

Index hasil dari fungsi, bukan nilai mentah.

```
INDEX ON users (lower(name))
```

Manfaat: Memungkinkan index dipakai saat melakukan query seperti `WHERE lower(name) = 'budi'`.

Kapan Kita Butuh Index?

✅ PAKAI INDEX JIKA

- Kolom sering muncul di klausa `WHERE`
- Kolom kunci untuk `JOIN` tabel
- Kolom dipakai untuk `ORDER BY` (Sorting)
- Kolom dipakai untuk `GROUP BY` atau Agregasi

❌ JANGAN PAKAI JIKA

- Tabel sangat kecil (scan biasa lebih cepat)
- Kolom sangat jarang digunakan untuk pencarian
- Data kolom tersebut sangat sering berubah/update (High Churn)
- Kardinalitas rendah (misal: kolom gender L/P)

Image Sources

https://www.smartsheet.com/sites/default/files/IC-Business-Emergency-Contact-Template.png

Source: www.smartsheet.com

https://people.iitism.ac.in/~sarun/notes/pho303/figs/pic3/c1.jpg

Source: people.iitism.ac.in