Бази даних

Лекція 14

Тематика лекції

- Pages, Tuples, MVCC
- Організація даних на диску
- Write-Ahead Log (WAL)

Кортеж (tuple)

Кортеж (tuple) - рядок даних таблиці в базі даних.

Кортеж на диску зберігається у наступному форматі:

- **метадані - tuple header (23 байта)**
- **бітмапа NULL значень колонок в даному рядочку**
- **дані - значення колонок в поточному рядочку**

Кортеж - заголовки

Основні частина заголовку кортежа:

- **xmin, xmax - id транзакцій, що створили та видалили кортеж**
- **ctid - посилання на місце зберігання кортежу або нової версії даного кортежу**
- **hoff (header offset) - адреса початку даних**
- **infomask - різноманітні прапорці**

MVCC

MVCC (Multi Version Concurency Control) - механізм, який використовує PostgreSQL для контролю конкурентних транзакцій. За своєю суттю це є Copy-on-Write.

Ідея MVCC полягає в тому, що кожний update - це запис нової версії рядочку. При цьому, транзакції, що почались до моменту update - можуть бачити лише стару версію (repeatable read).

MVCC

При оновленні рядочка створюється нова версія. Дані попередньої версії рядочка не змінюються, проте змінюються заголовки.

v1_tuple.xmax = transaction id
v2_tuple.xmin = transaction id
v2_tuple.xmax = 0
v1_tuple.ctid = v2_tuple

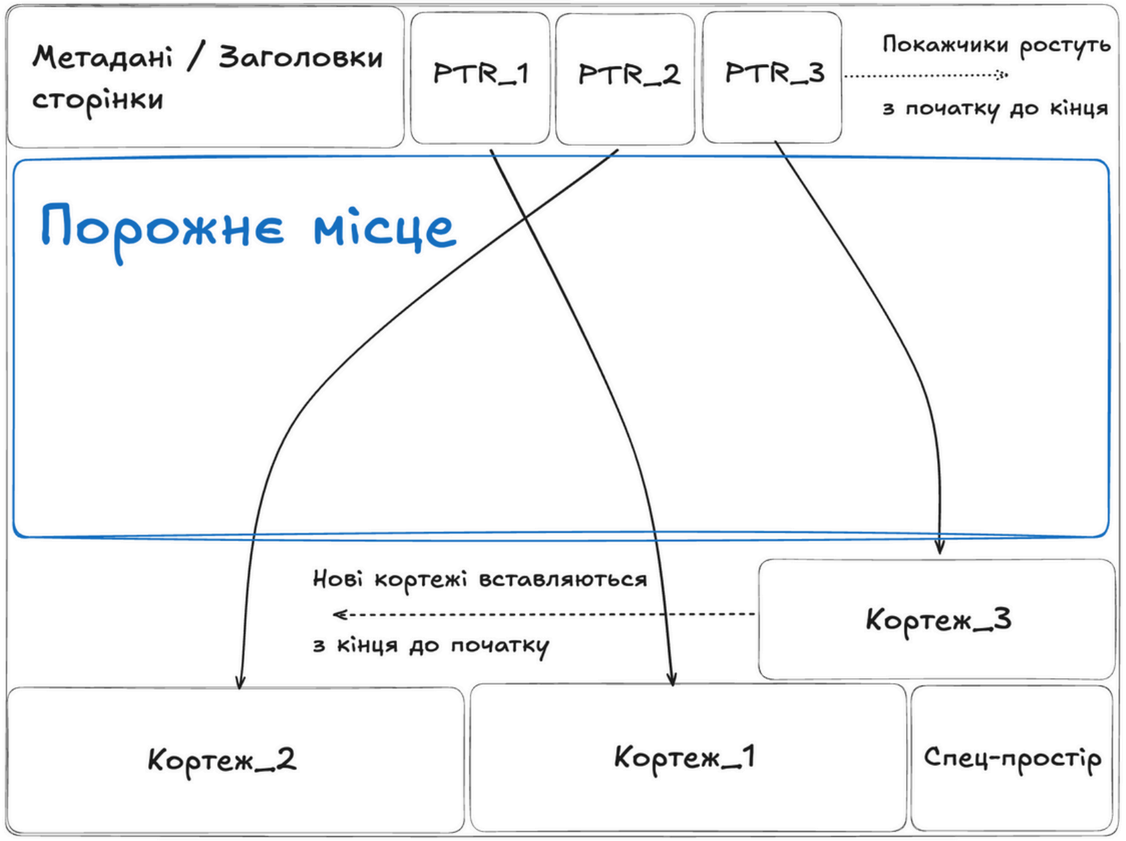
Сторінка (page)

Сторінка - логічна одиниця збереження даних базами даних на диску.

Сторінка - послідовний сектор пам’яті, що містить метадані та кортежі.

Сторінки завжди мають фіксований розмір (8KB за замовчуванням).

Структура сторінки



Заголовки сторінки

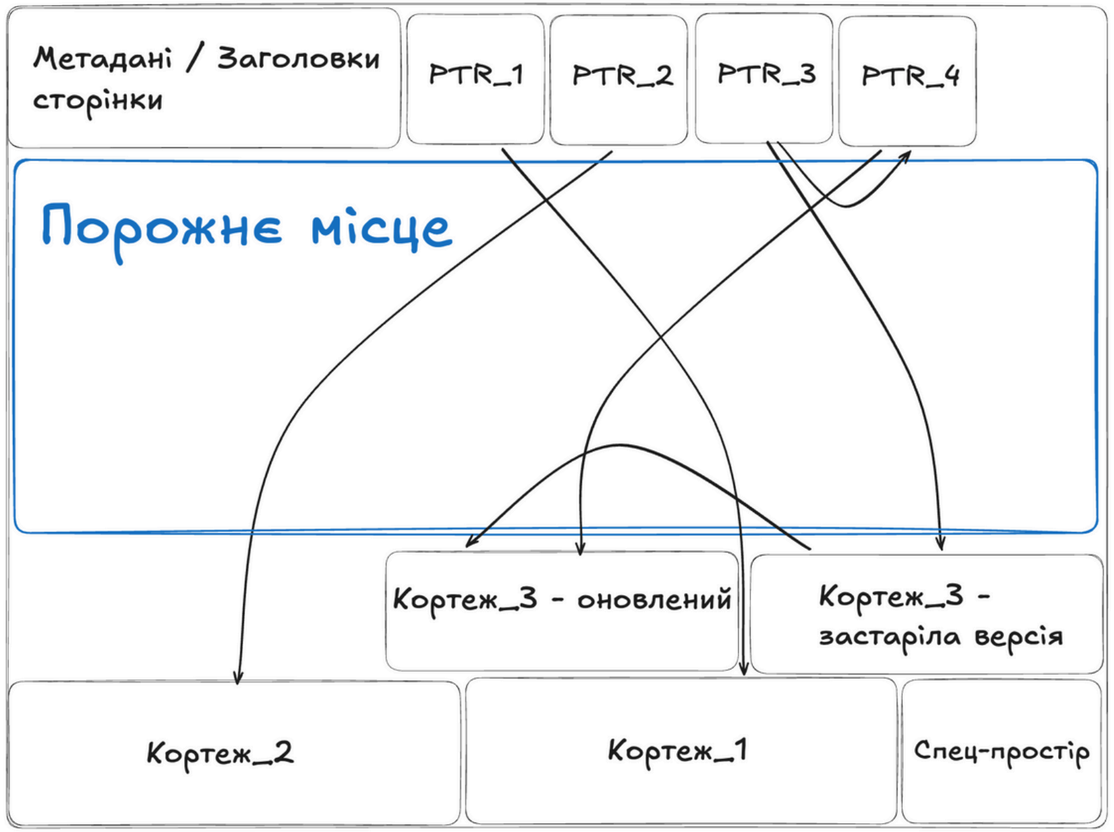
Заголовки мають розмір 24 байти та містять:

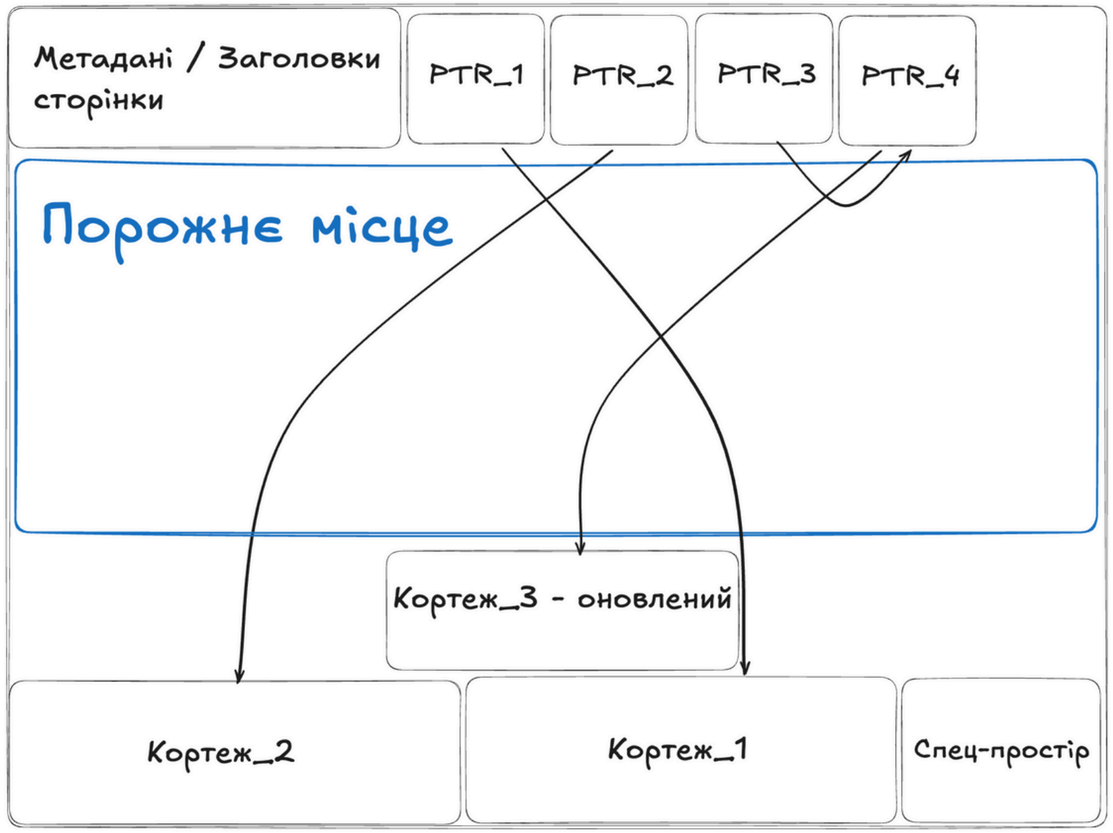
- **Чек-сума сторінки**
- **Посилання на початок і кінець вільного місця на сторінці**
- **Розмір сторінки та версія алгоритму роботи з сторінкою**
- **Найбільше значення xmax кортежів збережених на даній сторінці**

MVCC - hot update

Якщо при оновленні кортежа, нова версія зберігається на тій самі сторінці, що і попередня - оновлення індекса не відбувається, попередня версія просто зберігає посилання на нову версію. Тому це надзвичайно швидкий процес, який вимагає лише одного фізичного запису на диск.

MVCC - HOT update





Збереження сторінок

Сторінки зберігаються на диску у форматі кучі (Heap Files).

Куча (в даному контексті) - це неупорядкований набір сторінок.

Шлях до файлу, що зберігає сторінки:
$PGDATA/base/{id бази даних}/{id таблиці}

TOAST

Проблема: сторінки завжди мають фіксований розмір - 8KB.

```
CREATE TABLE test (
        id SERIAL,
        data TEXT
);
```

TOAST

Проблема: сторінки завжди мають фіксований розмір - 8KB.

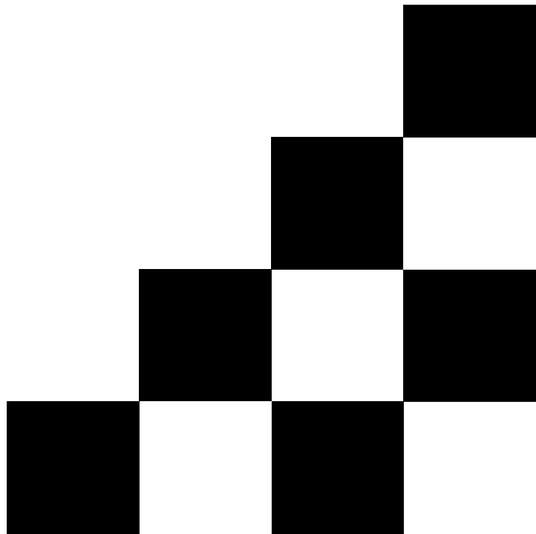
```
CREATE TABLE test (
        id SERIAL,
        data TEXT -- може містити значення більше 8KB
);
```

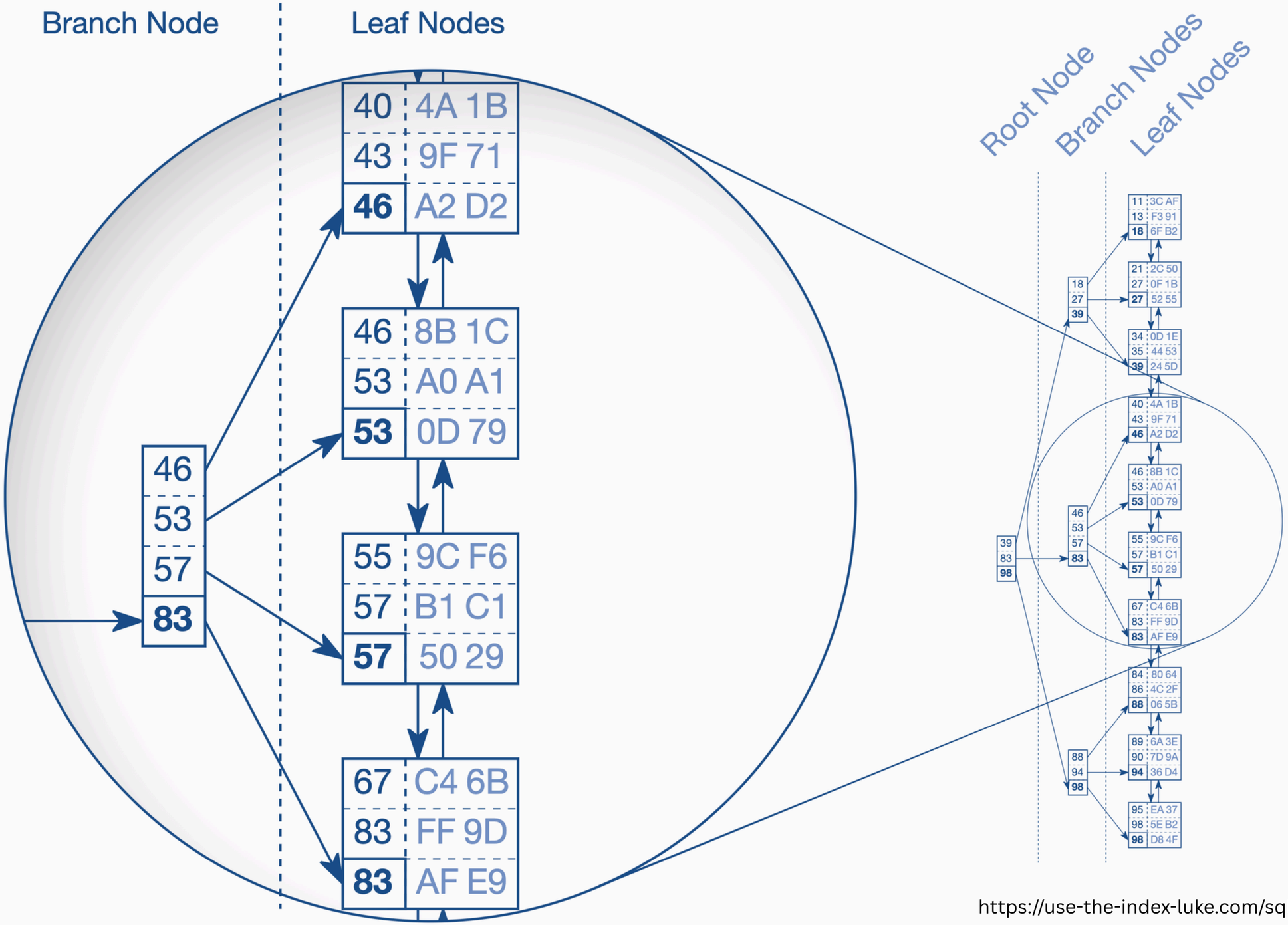
TOAST

Проблема: сторінки завжди мають фіксований розмір - 8KB.

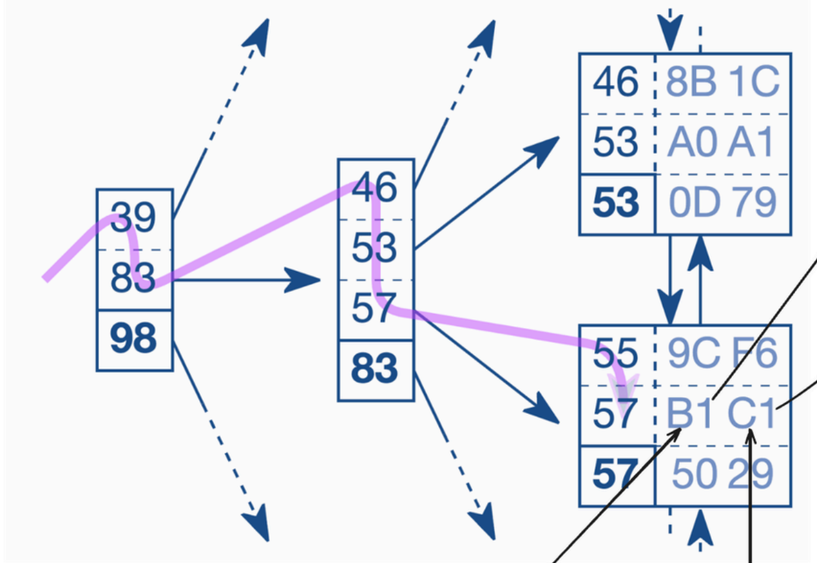
Рішення: The Oversized-Attribute Storage Technique (TOAST).
Ідея в тому, щоб великі значення зберігати в окремій таблиці, яка зберігає дані великого розміру, а основна таблиця зберігає лише посилання на великий “шматок” даних.

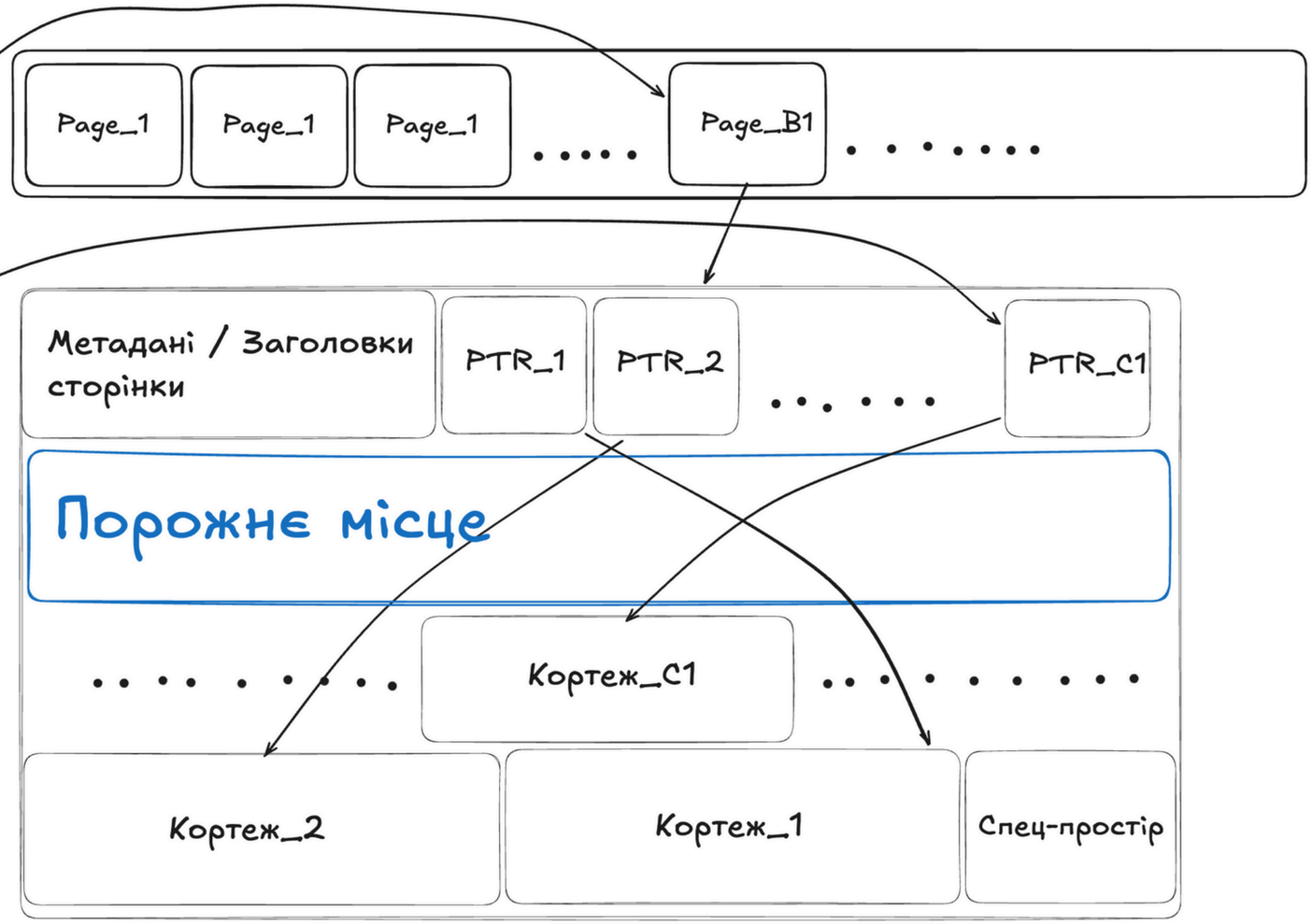
B-Tree





https://use-the-index-luke.com/sql/anatomy/the-tree





https://use-the-index-luke.com/sql/anatomy/the-tree

Збереження B-Tree на диску

B-Tree зберігається у вигляді сторінок по 8KB.
Кожна вершина дерева відповідає одній сторінці.
Кількість читань з диску для пошуку одного значення по індексу відповідає глибині дерева + 1 читання даних з сторінки таблиці.

Write-Ahead Log (WAL)

WAL - підхід, який використовує PostgreSQL для ефективного збереження транзакцій на диску.

Ідея WAL полягає в тому, що СУБД всі зміни в процесі транзакції записує у сторінки в оперативній пам’яті і також в оперативній пам’яті зберігає історію змін. При комміті транзакції відбувається запис історії змін на диск, а запис змінених сторінок відбувається пізніше у фоновому режимі.

Write-Ahead Log (WAL)

Основні принципи:

- **Лог змін записується перш за все.**
- **Коммiт транзакції очікує WAL fsync.**
- **Змінені сторінки записуються пізніше асинхронно.**
- **Періодично відбуваються чекпоінти - запис всіх сторінок з пам’яті на диск.**

Write-Ahead Log (WAL)

Переваги WAL:

- **Запис логу на диск значно швидший, ніж запис сторінок з даними.**
- **Тому запис логувідбувається синхронно, а запис сторінок - асинхронно.**
- **Якщо WAL записаний, а сторінки ні і стається збій - з логу можна відновити оригінальні дані.**

Висновки

- Все зберігається у форматі сторінок
- Сторінки завжди фіксованого розміру
- I/O операції з диском завжди по 8KB
- Дані таблиць зберігаються в рандомному порядку
- Всі рядочки на диску незмінні

Питання