

OLAP vs OLTP Databases

Online Transactional Processing (OLTP)

There are two major types of Relational Database architectures: OLTP and OLAP. OLTP stands for "Online Transactional Processing". This original architecture stemmed from early applications (like banking software) needing to process *rows* of information at a given time. You can picture this operation as if you are ordering something from Amazon. When you place your order, it will send a row of information that contains your name, address, credit card details, and special instructions, because it is all relevant to this single transaction. Having the database architecture setup this way makes reads and writes from the database for single transactional details (think row of data) lightning fast. So given the following table:

name	address	credit card number	special instructions
Mike	123 Sesame Street	1234-5678-9101-1123	Hazmat
Kelsey	456 Abc avenue	3211-1019-8765-4321	Fragile

You could query it like so:

```
SELECT
*
FROM orders
```

When the query engine retrieves the information from the database it will first retrieve Mike's row of information, then it will retrieve Kelsey's row of information, and so on until it reaches the end. This is extremely efficient if you are just needing to retrieve a single row of information's detail. Where this now becomes an issue is if you are wanting to retrieve *all* of the names of all the customers, then count how many there are in the table.

OLTP databases need to retrieve the entire row of information even if you are only wanting to select a single column. You can start to see where this becomes an issue if you are only looking to grab a single column and then aggregate that column. This becomes an even larger issue when there are dozens of columns in the table. As you can imagine, this can bog down your database's compute availability. Enter onto the scene the Online Analytical Processing database

Online Analytical Processing (OLAP)

OLAP databases (e.g. Redshift, Terradata) came to popularity because Business Intelligence employees were running queries to determine aggregate metrics like "*How many orders were placed last week?*" or "*What is our best selling product?*", but the query engines for OLTP engines (like MySQL or Postgres) are not optimized for this type of workload. As you recall, OLTP scans *rows* on disks, the main difference in OLAP, is that it scans *columns* on disk. This allows us to query over only the columns we need to aggregate rather than the whole row. So in or table above, if we just select the count of the names like so:

```
SELECT
count(name) as name_count
FROM orders
```

We are only going to scan one block of data (the single column) vs 2 blocks of data (2 rows). As you can imagine this becomes extremely efficient the more rows we have.

Keep in mind, I am over simplifying how this actually works for the sake of explanation and keeping the introduction to this concept *gentle*. While the data is stored in blocks like this, normally they have a size limit of 1mb on disk and then there are also indexes that are created which I will go into in a different post.

Revision #4

Created 14 June 2022 19:55:24 by Trevor Barnes

Updated 14 June 2022 20:25:52 by Trevor Barnes