OLTP refers to an Online transactional processing system

OLAP refers to an Online analytical processing system

In colloquial usage both a Data Warehouse and OLAP are the same. Though there's a subtle difference between both.

This is a good link which explains the deifferences between a data warehouse and an OLAP system: [https://technet.microsoft.com/en-us/library/aa1979...](https://technet.microsoft.com/en-us/library/aa197903) I found this [Stackoverflow question](http://stackoverflow.com/questions/18916682/data-warehouse-vs-olap-cube) to be useful too and that contradicts the below paragraph.

The difference is subtle because most Data warehouses today incorporate 'multidimensional data model and data aggregation techniques' within their schemas to support faster iterative querying of historical data.

Looks like OLAP refers to analysis of historical data while Data Warehousing refers to simple storage of historical data. Though in the industry the two are often used interchangeably because of how closely related the two concepts are.

Both types of systems (OLTP and OLAP) are implemented on a RDBMS or a relational database management system. Say, on a Oracle, MS SQL Server or Teradata system.

Whatever happens, at the end of the day, both an OLTP and an OLAP implemented on a RDBMS are basically CRUD or Create, Read Update, Delete systems.

So what's the difference between the two: OLTP and OLAP? If mostly they are the same. Well, the first major point of difference is in the schema or implementation of the databases.

OLAP schemas are optimized for writes and reads of a whole lot of records. On the order of millions and billions and even trillions. So, how do you this fast? By denormalizing the schema.

OLTP schemas on the other hand are optimized for Insertion, Update and Deletion of a few records, say at the max 10-20 or let's say even 100 or so records at a time. For such operations to work fast and work correctly, the schema needs to be normalized.

Now the question is why? **Why** do we need two different kinds of database systems? How are they even remotely useful, except for the purpose of creating new product categories for DB and BI vendors?

***Well, the why is best illustrated by an example and is the whole point of this blog post***.

**Consider**an **airline reservation system**, this is a transactional db or OLTP. When you book a reservation on such a system, a record is created for your reservation. Later on, if you make any changes to the reservation, your record is updated.

When you go to the airport and stand in line at the check-in counter, the check-in agent enters some unique key you had entered into the system at the time of reservation, such as passport number, phone number, or ticket number into the check-in system. And the system returns a single record corresponding to your reservation.

Now, assume I'm an analyst for the airline in question and I want to run a report and see what our most profitable routes are. And for this I need to access all the records and aggregate them by route. Also assume, this airline company is cheap and hasn't invested in an OLAP system.

So, I'd have to run this report on the same database as the one which holds your reservation. Now imagine that I'm running this report at the same time as you are standing in a line behind 20 other people at the airport.

***What happens next?***

Well, I go on sipping on my coffee waiting for my report and staring at the screen, while you my dear sir/lady are fretting in the line, glaring malevolently at the check-in agent. Meanwhile, the check in agent has no clue. And the airline's database has simply stopped processing requests from any of the check-in agents. It's just too busy running the friggin' profitability report. For it has to access millions and millions of rows, aggregate them, sort them and serve them up to me.

***Now, why would you want to fly with this airline again?***

So, that's the reason why corporations need a Data warehouse or an OLAP system (the terms are mostly interchangeable). See analysis is done on millions, billions, even trillions of records at once while transactions are usually carried one or a few records at a time. Hence, without separation between the two types of systems, a modern high velocity business would come to a stand-still.

**Let's look at a few more differences between the two:**

**OLTP**

*Records keep changing*, once the record is written, it keeps getting updated based on user input. Say, you made a hotel booking and after a few days need to change it, the record for your original booking would be updated. It won't be created anew.

*Updates are not kept track of*. Say, if you change the date today for your booking, you can't tell what your original booked date was by looking into the database. Unless of course, the schema is designed such that a history of the records is maintained within the database.

*They are normalized to the 3NF* or higher, so that any updates to a record are applied to only a single table. Since we keep making a lot of updates, it's necessary to keep the tables touched by the update to a minimum.

**OLAP**

*Records usually don't change*once they are loaded into an OLAP. They are time stamped and when people read from an OLAP they read by filtering on date.

*Updates are very rare*on records in a DW or OLAP. They are read for reporting purposes. Millions and billions of records at a time, processed by whatever process needs to be applied. Usually aggregation in terms of sum, counts, average et., al.

*DWs are not usually normalized* in order to speed up the read and write times.

*The dominant architecture is the star-join*. Visualize a single or say two or three fact tables in the center around which the dimension tables are organized, and the dimension tables connect to the fact tables but not usually other dimension tables, this forms the star-join architecture of a Data warehouse or OLAP system.

*— FEB 15, 2015 —*