What's a data warehouse?

Where warehouse workers means something else

Justin
Aug 4, 2020

The TL;DR

A data warehouse is a special type of database designed for **analytics instead of transactions**.

- There are two types of database use cases: **transactional** and **analytical**
- Transactional databases like PostgreSQL are built for **adding, updating, and removing data**, but analytical DBs are for **complex queries and joins**
- Data gets moved from transactional to analytical databases through **a process called ETL**

Data warehouses are one of the fastest growing segments in cloud, and power most of what's happening in modern data science. So read this!

Transactional vs. analytical databases

> 🔮 **Dependencies** 🔮
>
> To get the most out of this post, you'll need to understand what a database is, how schemas and relations work, and how software runs in the cloud.
>
> 🔮 **Dependencies** 🔮

The core of whatever app you're reading this in is a production database: it stores user information and other backend stuff that populates whatever you see on the screen (unless you printed this out, loser). That database is all about *transactions*:

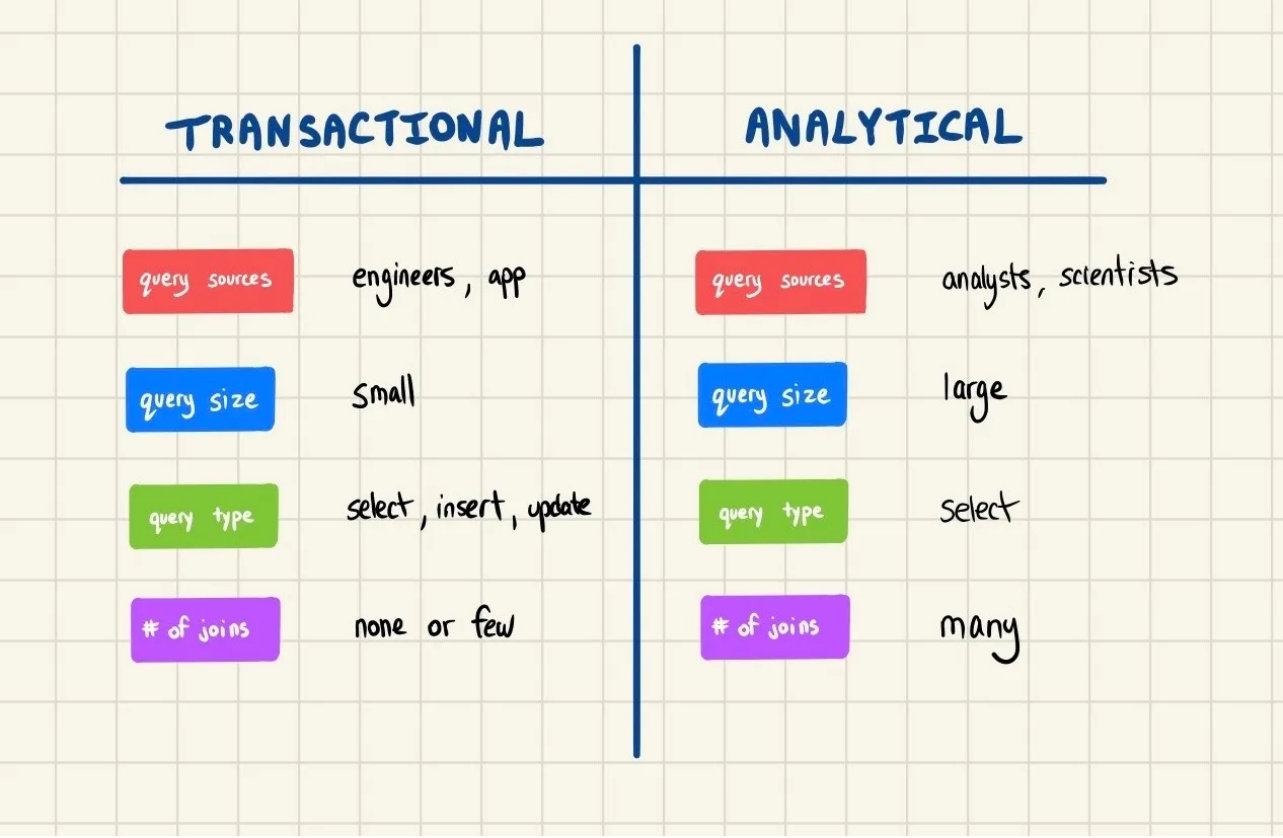
- Every time a new user signs up a new row gets **added**
- If you change your password, your row gets **updated**
- If you shut down your account, your row gets **deleted** (sometimes)

These kinds of processes and how they interact with a database are called **OLTP**, or online transactional processes. Scary acronyms!

Because a lot of these *transactions* can be happening at once (apps with many users or page loads), popular databases like MySQL are built with that in mind. They usually have special features that make sure these transactions don't get messed up, and optimize for that kind of use case. It's all about transaction after transaction, boom boom boom, making sure your app keeps working without getting messed up.

You can think of OLTP operations as text messages. It's a great medium for short, clear directives. Need a quick favor. What's Jason's number? Are you going to the party tonight? Texts are about speed and efficiency.

There's a whole other use case for databases though: **analytics**. If you have *big questions* you want to ask of your data, like how many users you have, how many orders get cancelled each month, or how much money you made last year, you use the database in a very different way. Analytical queries typically require looking at a lot more data, take a much longer time to run, and join a bunch of different sources together. The process is usually called **OLAP**, or online analytical processing.



These kinds of analytical transactions are like email: it's a much better format for longer form, more involved communication than texts are. Emails take longer to send, but they're more organized and professional for when you need to send multiple paragraphs and embedded media.

Data warehouses and ETL

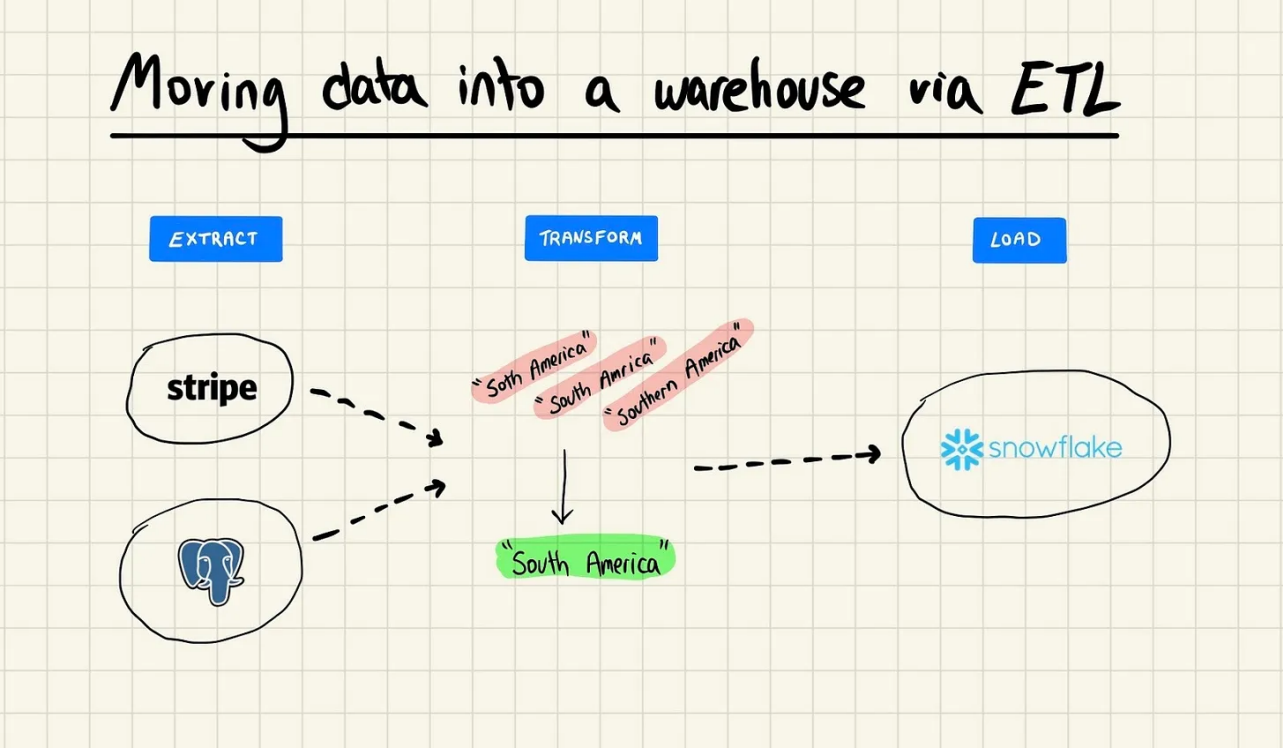
Because there's such a big difference between *transactional* and *analytical* use cases, they require different tools; just like a bike and a car are both great, but for very different situations. A data warehouse is just a group of databases built for *analytics* instead of *transactions*:

- Warehouses bring together data from **lots of sources** into one place
- Apps rarely interact directly with warehouses: they're mostly used by **analysts and data scientists**
- Data warehouses usually have **a lot more data** in them than transactional DBs

The data that companies put in their warehouses is usually a combination and/or transformation of data that they have in other systems. The cool thing about the warehouse is that it brings those all together. Example time!

Let's say I sell sneakers on an e-commerce site, where users can create and log in to accounts. I have a production, transactional database with information about my users, as well as some Stripe data about payments and orders. I want to build a dashboard that shows **how much money I make monthly from users in South America**. How do I do that? I'll:

- Clean up my user data to make sure that mistakes like "Soth America" get interpreted correctly (transform)
- Pull together my user data and my Stripe data so I can filter for users in South America (combine)
- Run a query every day that calculates monthly revenue, and stores it in a table in my warehouse



The end result is a new data set that has the information I need, stored in a data warehouse. This process that we just went through is called **ETL** – an acronym for **extract**, **transform**, **load** – and it's how we take data out of systems, do stuff with it, and then put it into the warehouse. Sometimes the order gets reversed (ELT), but that's for another time.

> 🚨 **Confusion Alert** 🚨
>
> One thing that always tripped me up about ETL and Warehouses: sometimes, the process and destination are *really simple*. ETL doesn't need to mean some complex pipeline: even just taking data to calculate and store monthly averages counts too.
>
> 🚨 **Confusion Alert** 🚨

Because data warehouses are used so differently than transactional DBs, they usually run on different infrastructure. Data warehousing providers need to deal with storing petabytes of data and running special kinds of queries, as well as bundling ETL related features. A popular one is Snowflake, who was in the news recently for being valued at $12B, or the average price of a house in Palo Alto.

"Data warehouse" in conversation

“You really shouldn’t query the production database like that, use the warehouse”

If you’re writing a big query with a lot of joins, don’t bog down the transactional database: use the tables we put in the data warehouse.

“We can create an ETL pipeline to get that data ready for you”

You can’t get the data you need from our production systems, but we can build a job that gets it all together in the warehouse for you.

“We’re running out of space in our RDS instance, we should probably move over to Redshift”

Instead of storing our warehouse data in a system built for transactions, we should move it over to a purpose built warehouse solution like AWS Redshift.

Terms and concepts covered

```
Transactional, analytical


OLTP


OLAP


Data warehouse


ETL
```

Further reading

- As analytics and data science are having their moment, the infrastructure powering data warehousing is booming: popular provider Snowflake is now worth $12B
- Mature companies can have hundreds or thousands of ETL jobs running daily. Open source software like Apache Airflow helps manage that
- As data storage has gotten cheaper, dev teams have started loading data into the warehouse and *then* transforming it, which is called ELT

2 Comments

Write a comment...

Tuan-Anh Sep 16, 2021

would love to read your post about data warehouse schema design!

♡ 4 Reply Gift a subscription Collapse •••

Wes May 21, 2021

Thank you for this--I just came back to it after a few weeks. It was *exactly* what I needed to clarify the difference between how Postgres and BigQuery were used.

♡ 2 Reply Collapse •••