















What is Data Engineering:

Data Engineering comprises all engineering and operational tasks required to make data available for analytics including but not limited to: Data Ingestion, Data Synchronization, Data Transformation, Data Models, Data Governance, Performance Optimization, Production orchestration

Data engineer always works on retrieving the data and managing the data. It is essential to know how to handle the huge amount of data. Data can be in any form structured, unstructured, log files, flat files and web scraped data etc. How to handle all sort of data? And how we can convert this raw data into useful information to get insights from the data (Which can be further processed by a Data analyst or a Data Scientist).

Data Scientist also known as Data Managers, statisticians.	Data Engineers also known as database administrators and data architects.	Data Analysts also known as business Analysts.
		
A data scientist will be able to take data science projects from end to end. They can help store large amounts of data, create predictive modelling processes and present the findings.	They are versatile generalists who use computer science to help process large datasets. They typically focus on coding, cleaning up data sets, and implementing requests that come from data scientists.	They typically help people from across the company understand specific queries with charts.
Skills: Mathematics, Programming, Communication	Skills: Programming, Mathematics, Big data	Skills: Statistics, Communication, Business knowledge
  	  	  

What Data Engineers do:

- Architect distributed systems
- Creating reliable pipelines
- Combining data sources
- Architect data stores
- Collaborating with data science teams and building the right solutions for them

Data Engineers Key skills:

Since Data Engineers are much more concerned with analytics infrastructure, most of their required skills are, predictably, architecture-centric.

In-depth knowledge of SQL and other database solutions

Data Engineers need to understand database management, and as such, in-depth knowledge of SQL is hugely valuable. Likewise, other database solutions, such as Cassandra or Bigtable, are great to know if you plan on doing freelance or for hire engineering, as not every database is going to be built in the recognizable standard.

Data warehouse architecture and ETL tools



Data warehousing and ETL experience is essential to this position. Data warehousing solutions like Redshift or Panoply, as well as familiarity with ETL Tools, such as with StitchData or Segment is hugely valuable. Similarly, experience with data storage and retrieval is equally vital, as the amount of data being dealt with is simply astronomical.

Hadoop based Analytics (Hbase, Hive, Mapreduce etc)

Strong understanding of apache Hadoop-based analytics are very common requirements in this space, with knowledge of Hbase, Hive, and Mapreduce often considered a requirement.

Coding

Speaking of solutions, knowledge of coding is a definite plus here (and also possibly a requirement for many positions). Familiarity, if not outright expertness, is very valuable in Python, C/C++, Java or other such languages.

Machine Learning

While mainly the focus of data scientist, some level of understanding of how to act upon this data is also invaluable for Data Engineers. For this reason, some knowledge of statistical analysis and the basics data modeling are hugely valuable.

While machine learning is technically something relegated to the Data Scientist, knowledge in this area is helpful to construct solutions usable by your cohorts. This knowledge has the added benefit of making you extremely marketable in this space, as being able to “put on both hats” in this case makes you a formidable tool.

Data Engineering Challenges:

- Data Integrity problems
- Often changing of Schema
- In small organizations, there wont be proper data infrastructure to maintain the data and the role also may be overloaded around setting up and operating the organization’s data infrastructure.
- In large organizations, where hundreds of people are involved in the data generation side of analytical process, consensus seeking is challenging, when not outright impossible in a timely fashion.
- ETL jobs takes long time and it gives boredom and lead to unhealthy work life balance
- Lack of technical skills to perform day to day operations
- Lots of tools are available to use and having knowledge on every tool is impossible

Here are some of the concepts which are useful to handle data:



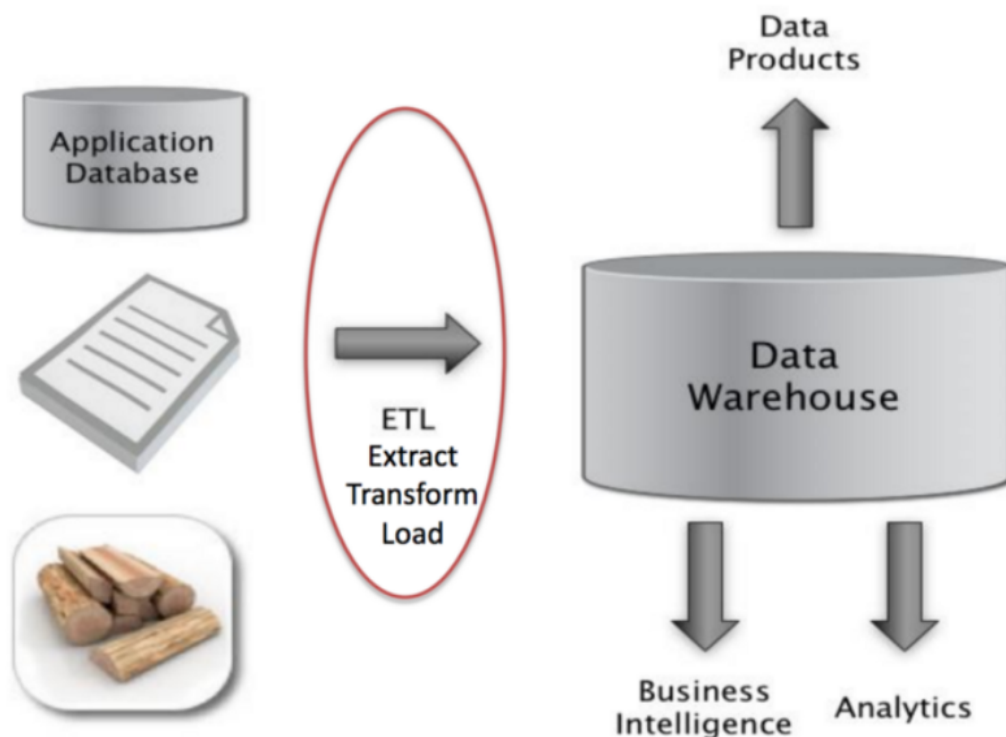
- Database
- Data Warehouse
- Data Lake
- ETL

Database: Database is a systematic collection of data. Databases support storage and manipulation of data.

Data Warehouse : It is a blend of technologies and components which allows the strategic use of data. It is a technique for collecting and managing data from different sources.

It stores large amount of information by a business which is designed for query and analysis instead of transaction processing. Data warehouse is mostly useful for Business Intelligence because it handles structured data and processing time will be less as it transforms the data before loading into data base. Data warehouse use the concept of extract, transform and loading technique(ETL).

The Big Picture

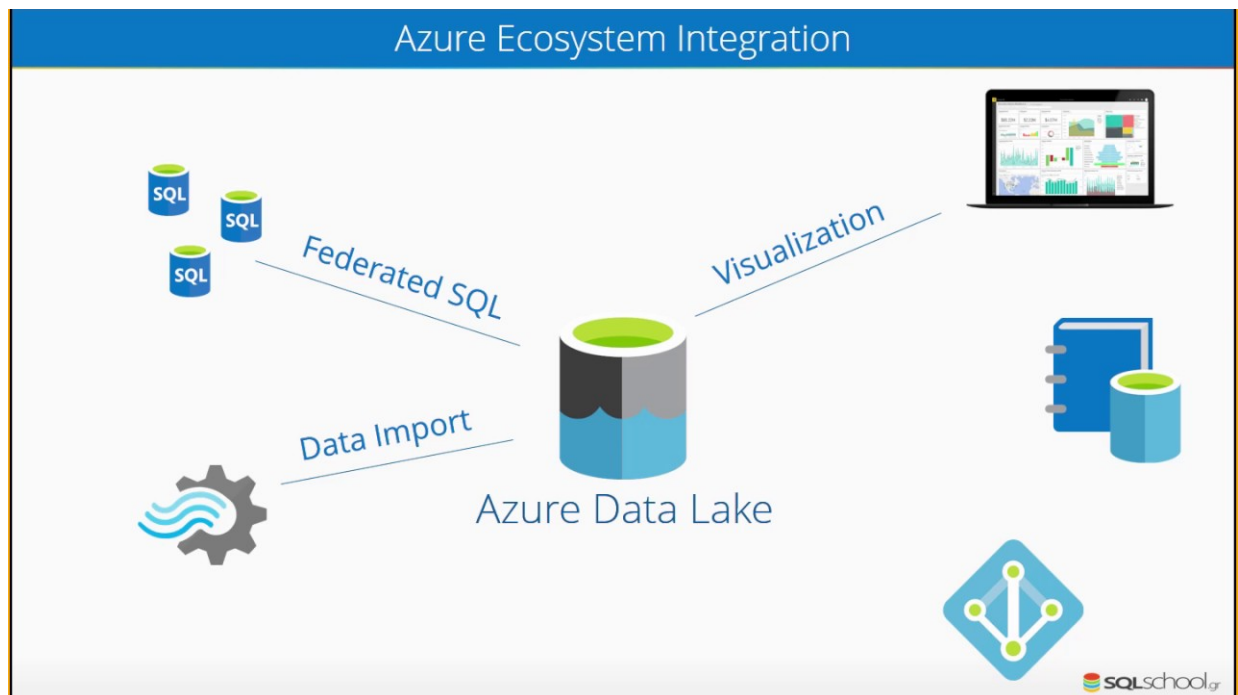


Data Lake : It is a storage repository that can store large amount of structured, unstructured and semi structured data. It is a place to store every type of data in its original format with no limits on account size or file. It offers high data quantity to increase analytic performance and native integration.



The data which stored in data lake can be transformed based on business needs, until that it will be in raw format.

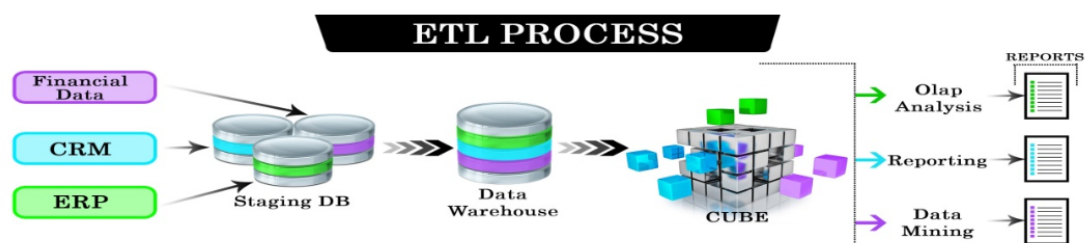
Data lake use the concept of extract, load and transform technique(ELT).



ETL : ETL is an abbreviation of Extract, Transform and Load. In this process, an ETL tool extracts the data from different RDBMS source systems then transforms the data like applying calculations, concatenations, etc. and then load the data into the Data Warehouse system.

It's tempting to think a creating a Data warehouse is simply extracting data from multiple sources and loading into database of a Data warehouse. This is far from the truth and requires a complex ETL process. The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

In order to maintain its value as a tool for decision-makers, Data warehouse system needs to change with business changes. ETL is a recurring activity (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



What we gonna cover in Beginner Data Engineering Track:

SQL concepts using MySQL

Basics of No SQL

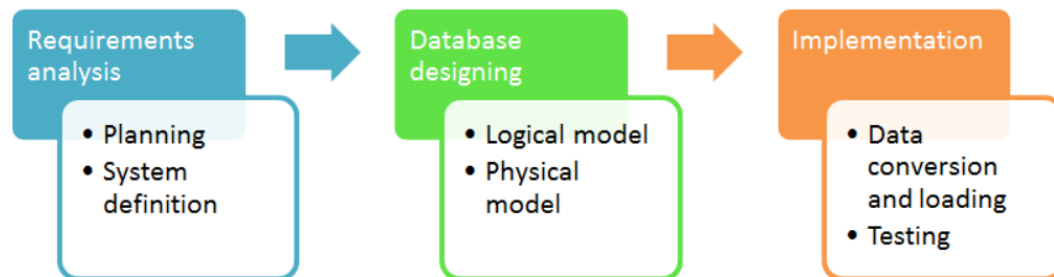


Google BigQuery - Cloud platform

Databases: In simple terms,

- It is a collection of data
- Are highly structured data files
- Allow data input , organization and retrieval
- Use tables to store ,sort and filter the data.

Database development life cycle



Requirements analysis

- **Planning** - This stage concerns with planning of entire Database Development Life Cycle. It takes into consideration the Information Systems strategy of the organization.
- **System definition** - This stage defines the scope and boundaries of the proposed database system.

Database designing

- **Logical model** - This stage is concerned with developing a database model based on requirements. The entire design is on paper without any physical implementations or specific DBMS considerations.
- **Physical model** - This stage implements the logical model of the database taking into account the DBMS and physical implementation factors.

Implementation

- **Data conversion and loading** - this stage is concerned with importing and converting data from the old system into the new database.
- **Testing** - this stage is concerned with the identification of errors in the newly implemented system. It checks the database against requirement specifications.

Database Management System(DBMS):

Program that allows the manipulation of our database files and controls the hardware that stores them.

- It helps to create structural rules to store the data
- We can input the data based on the structural rules which we created before
- It helps to perform administrative works to protect the data



Database requirements:

- Data integrity
- Supported operations
- Modeling relationships
- Supported data types
- Transaction support
- Scalability
- Redundancy
- Schema flexibility
- Performance
- Integrations

Types of databases available:

RDBMS(Relational Database Management System)

Document databases

Columnar databases

Key-value databases

Graph databases

Hadoop Distributed File System(HDFS)

RDBMS:

RDBMS Characteristics

Product	Offers	Conv	Conv %
HDMI Cable	726	128	17.6%
Headset	690	132	19.1%
Keyboard	724	128	17.7%
Lightning Cable	722	144	19.9%
Mouse	715	140	19.6%
USB Drive	710	139	19.6%

- Tables, rows, columns
- Schema driven
- SQL-based queries
- Number crunching

Popular Products of RDBMS

Oracle

Microsoft SQL Server

MySQL

PostgreSQL

Microsoft SQL Server:



Strengths	Shortcomings	Applications
<ul style="list-style-type: none">• Transactions• Speed• Data integrity• Security	<ul style="list-style-type: none">• Linear scaling• Text/media• Complex data types• Cost	<ul style="list-style-type: none">• Master data• OLTP• Update intensive data

Document Databases:

Also called as NoSQL database

Document Database Characteristics

- JSON documents
- Flexible structures
- Document key
- Indexing
- Sharded clusters

```
{  
  "name": "Mike",  
  "age": 40,  
  "married": "yes",  
  "contacts" : {  
    "phone": 1234556,  
    "email": "m1@gm.com"  
  }  
}
```

Popular Products of Document Databases:

MongoDB
Elasticsearch
Couchbase

MongoDB:

Strengths	Shortcomings	Applications
<ul style="list-style-type: none">• Complex data types• Rich querying• Full text search• Scalability	<ul style="list-style-type: none">• No transactions• No media storage• Joins not optimal• No attribute-based security	<ul style="list-style-type: none">• Blogs• Catalogs• RDBMS alternative• Searchable repository

Columnar Databases:



Columnar Database Characteristics

Product	Offers	Conv	Conv %
HDMI Cable	726	128	17.6%
Headset	690	132	19.1%
Keyboard	724	128	17.7%
Lightning Cable	722	144	19.9%
Mouse	715	140	19.6%
USB Drive	710	139	19.6%
USB Hub	713	142	19.9%

- Stored column by column
- Tables/column families
- Key driven
- Supports multiple data types
- Empty column data not stored

Popular Products of Columnar Databases:

Cassandra

HBase

Google Bigtable

Cassandra:

Strengths	Shortcomings	Applications
<ul style="list-style-type: none">• CQL – like SQL• Optimal updates• Transactions• Scalability	<ul style="list-style-type: none">• No joins• Query by key only• No order by• Only small blobs	<ul style="list-style-type: none">• Data warehouse• Real-time counters• Customer 360• Write intensive apps• Machine learning

Key-value Databases:



Key-Value Database Characteristics

Key	Value
as12B5	10001
Emp-10023-Name	Mike Smith
Emp-10023-Qual	['BS', 'MS']
Cart-01999282	['Keyboard']

- Keys and values
- Access by keys
- Composite keys
- Memory caching
- Distributed

Popular Products of Key-value Databases:

Redis

Memcached

Hazelcast

Redis:

Strengths	Shortcomings	Applications
<ul style="list-style-type: none">• Ultrafast access• Rich data types• Auto-expire keys• Scalability	<ul style="list-style-type: none">• No search on values• Small values only• Consistency across nodes• No tables	<ul style="list-style-type: none">• Session cache• Shopping cart• Scorecard• Real-time queue

Graph Databases:



Graph Database Characteristics



- All about relationships
- Nodes and attributes
- Relationships and attributes
- Query by relationship
- Chaining

Popular Products of Graph Databases:

Neo4j
Giraph
TigerGraph

HDFS:

HDFS Characteristics

- Files and directories
 - Distributed file system
 - Any type of file
 - Commodity hardware
 - Streaming access
 - Query support with Hive and Impala
-



Strengths	Shortcomings	Applications
<ul style="list-style-type: none">• Linear scaling• Redundancy• Security• High availability	<ul style="list-style-type: none">• No updates• Limited querying• Queries – very slow• Not for small data	<ul style="list-style-type: none">• Raw dumps• Media storage• Data backups