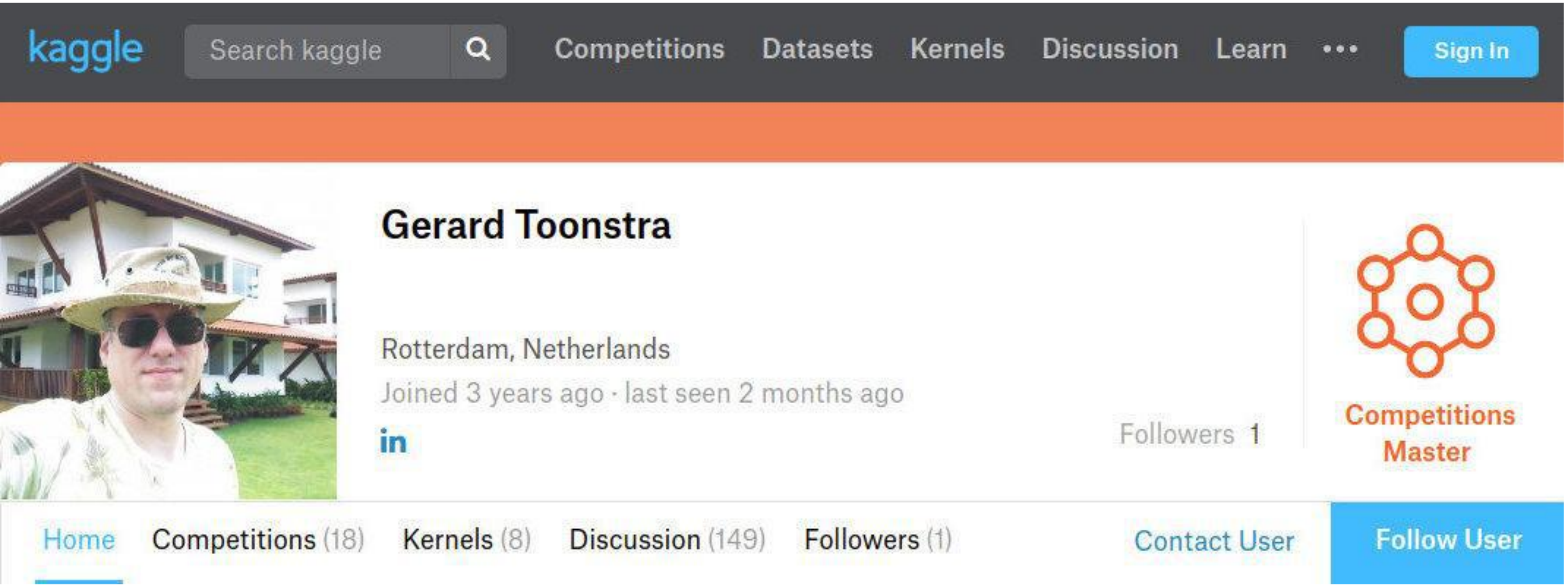
Big Data Vilnius

Agile Data Architecture

bigdata REPUBLIC

About me



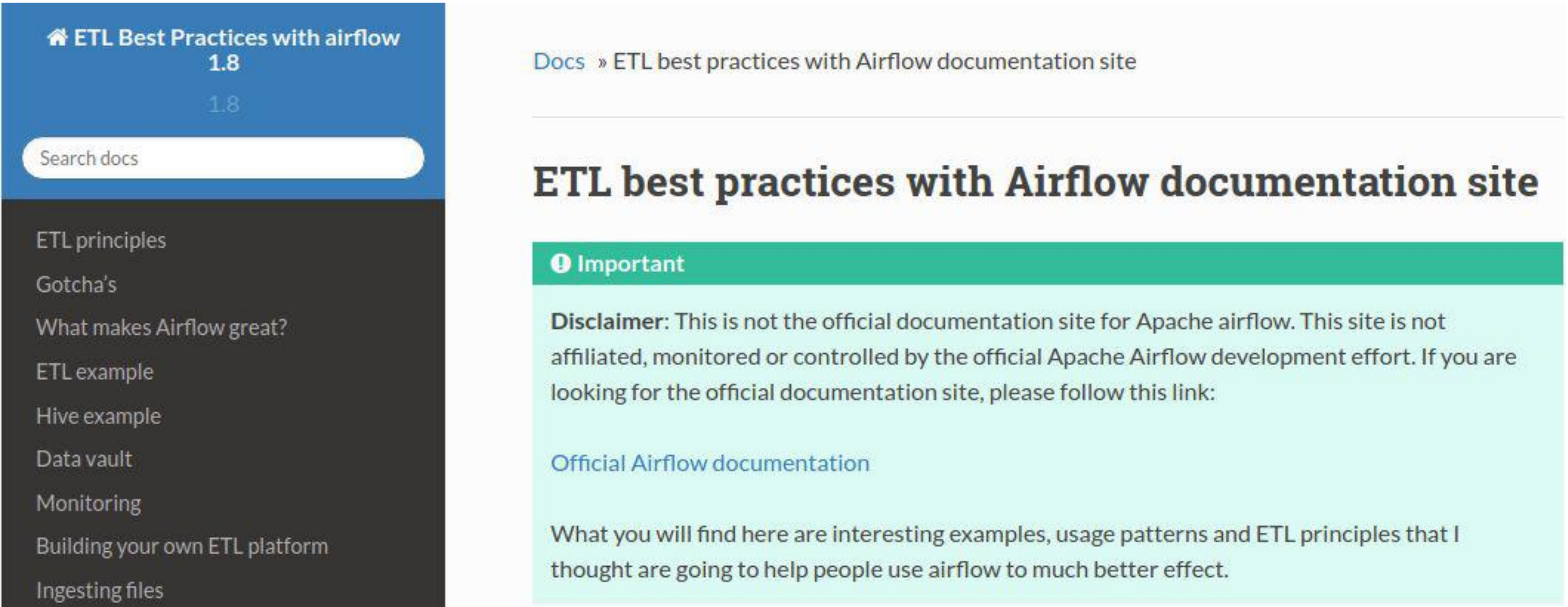
Gerard Toonstra

Rotterdam, Netherlands

Joined 3 years ago · last seen 2 months ago

Followers 1

Competitions Master


ETL Best Practices with airflow 1.8
1.8
Search docs
ETL principles
Gotcha's
What makes Airflow great?
ETL example
Hive example
Data vault
Monitoring
Building your own ETL platform
Ingesting files
Docs » ETL best practices with Airflow documentation site

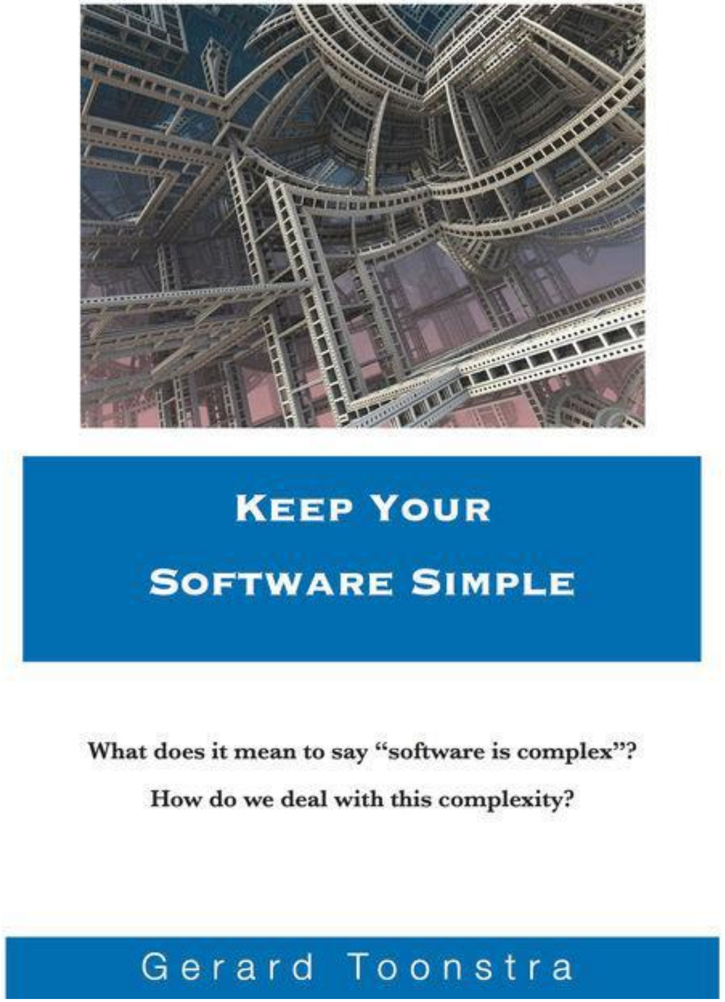
ETL best practices with Airflow documentation site

Important

Disclaimer: This is not the official documentation site for Apache airflow. This site is not affiliated, monitored or controlled by the official Apache Airflow development effort. If you are looking for the official documentation site, please follow this link:

Official Airflow documentation

What you will find here are interesting examples, usage patterns and ETL principles that I thought are going to help people use airflow to much better effect.



KEEP YOUR SOFTWARE SIMPLE

What does it mean to say "software is complex"?

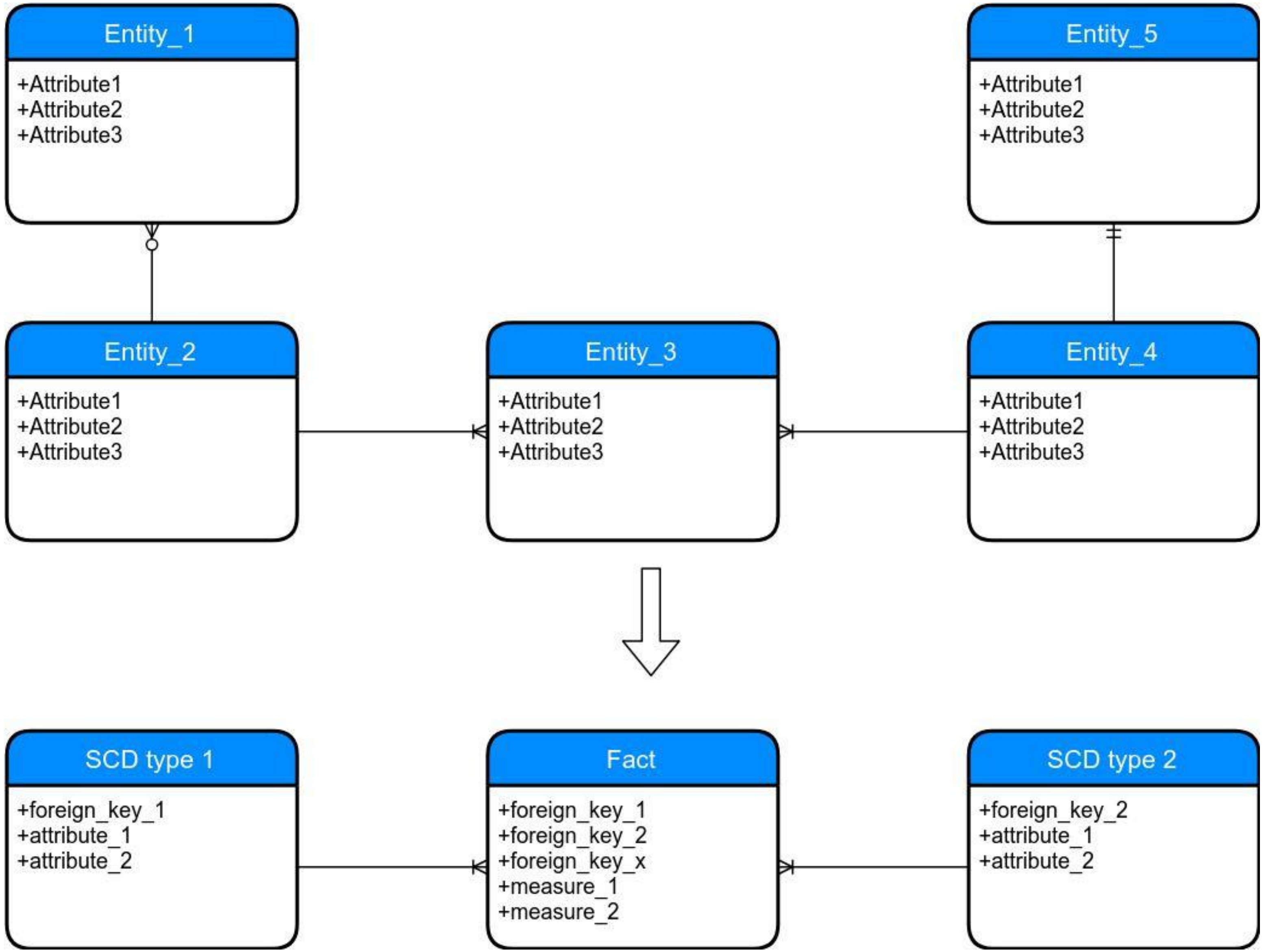
How do we deal with this complexity?

Gerard Toonstra

bigdata REPUBLIC

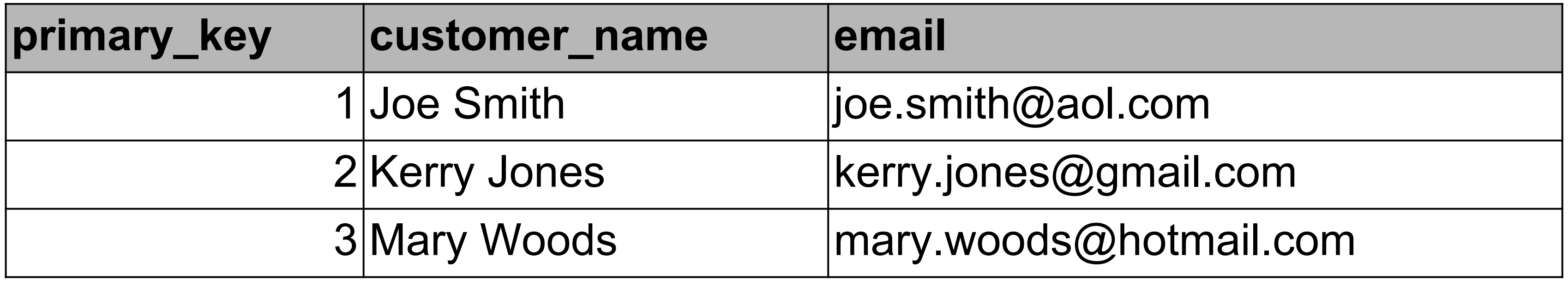
What is an agile data warehouse?

Design “lock-in”



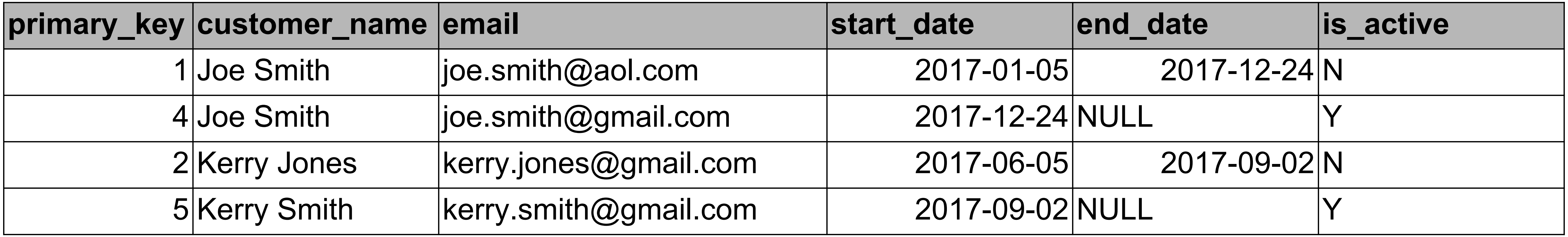
Design “lock-in”

Slowly Changing Dimension Type 1



primary_key	customer_name	email
1	Joe Smith	joe.smith@aol.com
2	Kerry Jones	kerry.jones@gmail.com
3	Mary Woods	mary.woods@hotmail.com

Slowly Changing Dimension Type 2



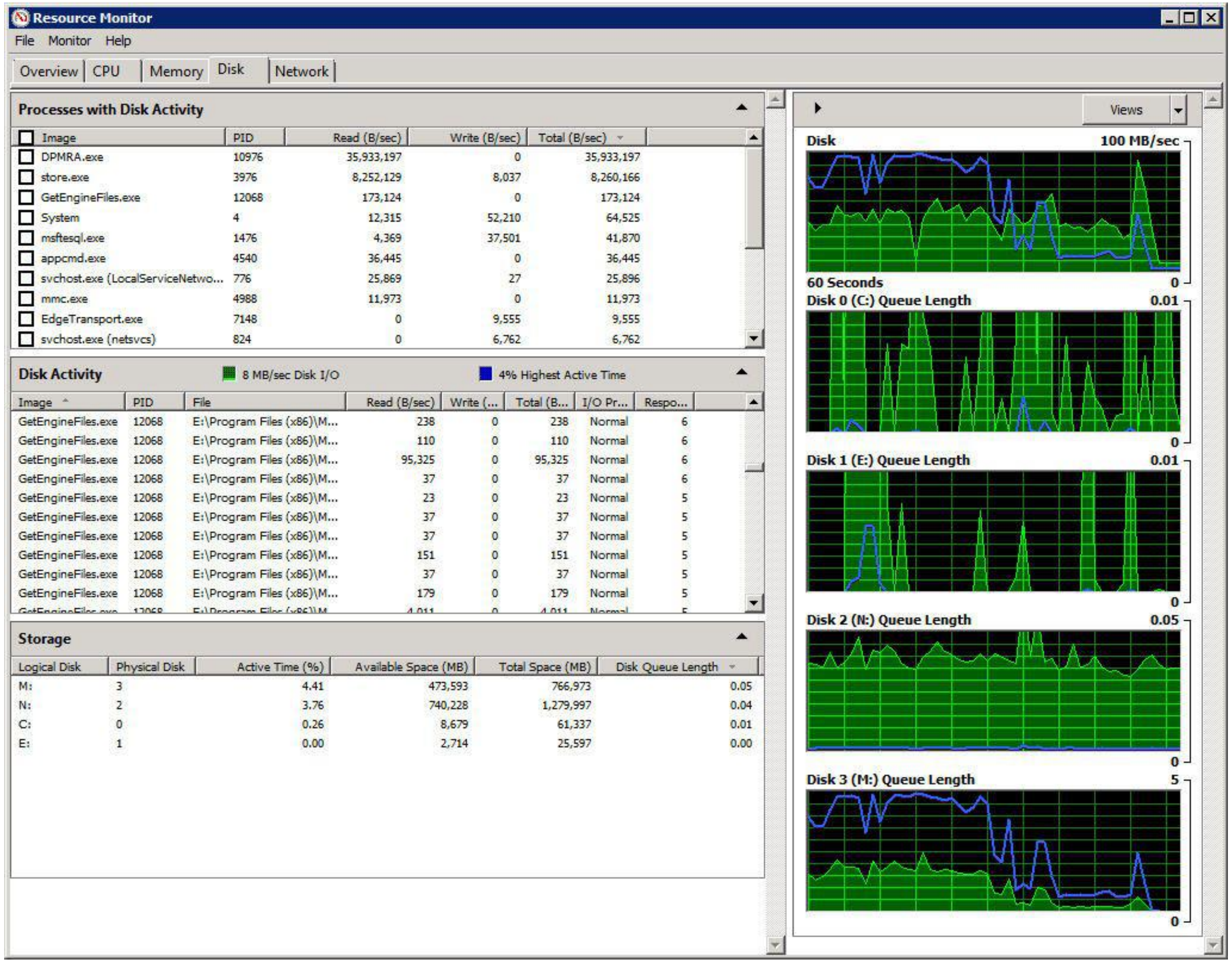
primary_key	customer_name	email	start_date	end_date	is_active
1	Joe Smith	joe.smith@aol.com	2017-01-05	2017-12-24	N
4	Joe Smith	joe.smith@gmail.com	2017-12-24	NULL	Y
2	Kerry Jones	kerry.jones@gmail.com	2017-06-05	2017-09-02	N
5	Kerry Smith	kerry.smith@gmail.com	2017-09-02	NULL	Y

Complexity: doing too many things at once

```
INSERT INTO customer_dim
SELECT source_cust_id, first_name, last_name, eff_date, end_date, current_flag
FROM
( MERGE customer_dim cm
  USING customer_source cs
  ON cm.source_cust_id = cs.source_cust_id
  WHEN NOT MATCHED THEN
    INSERT VALUES (cs.source_cust_id, cs.first_name, cs.last_name, convert(char(10), getdate()-1, 101),
’12/31/2199′, ‘y’)
  WHEN MATCHED AND cm.current_flag = ‘y’ and cm.last_name <> cs.last_name THEN
    UPDATE SET cm.current_flag = ‘n’, cm.end_date = convert(char(10), getdate()- 2, 101)
  OUTPUT $Action action_out, cs.source_cust_id, cs.first_name, cs.last_name, convert(char(10),
getdate()-1, 101) eff_date, ’12/31/2199′ end_date, ‘y’ current_flag
) AS merge_out
WHERE merge_out.action_out = ‘UPDATE’;
```

Data volumes

- Failure to meet SLA



Processes with Disk Activity

□ Image	PID	Read (B/sec)	Write (B/sec)	Total (B/sec)
□ DPMRA.exe	10976	35,933,197	0	35,933,197
□ store.exe	3976	8,252,129	8,037	8,260,166
□ GetEngineFiles.exe	12068	173,124	0	173,124
□ System	4	12,315	52,210	64,525
□ msftesql.exe	1476	4,369	37,501	41,870
□ appcmd.exe	4540	36,445	0	36,445
□ svchost.exe (LocalServiceNetwo...	776	25,869	27	25,896
□ mmc.exe	4988	11,973	0	11,973
□ EdgeTransport.exe	7148	0	9,555	9,555
□ svchost.exe (netsvcs)	824	0	6,762	6,762

Disk Activity 🟩 8 MB/sec Disk I/O 🟦 4% Highest Active Time

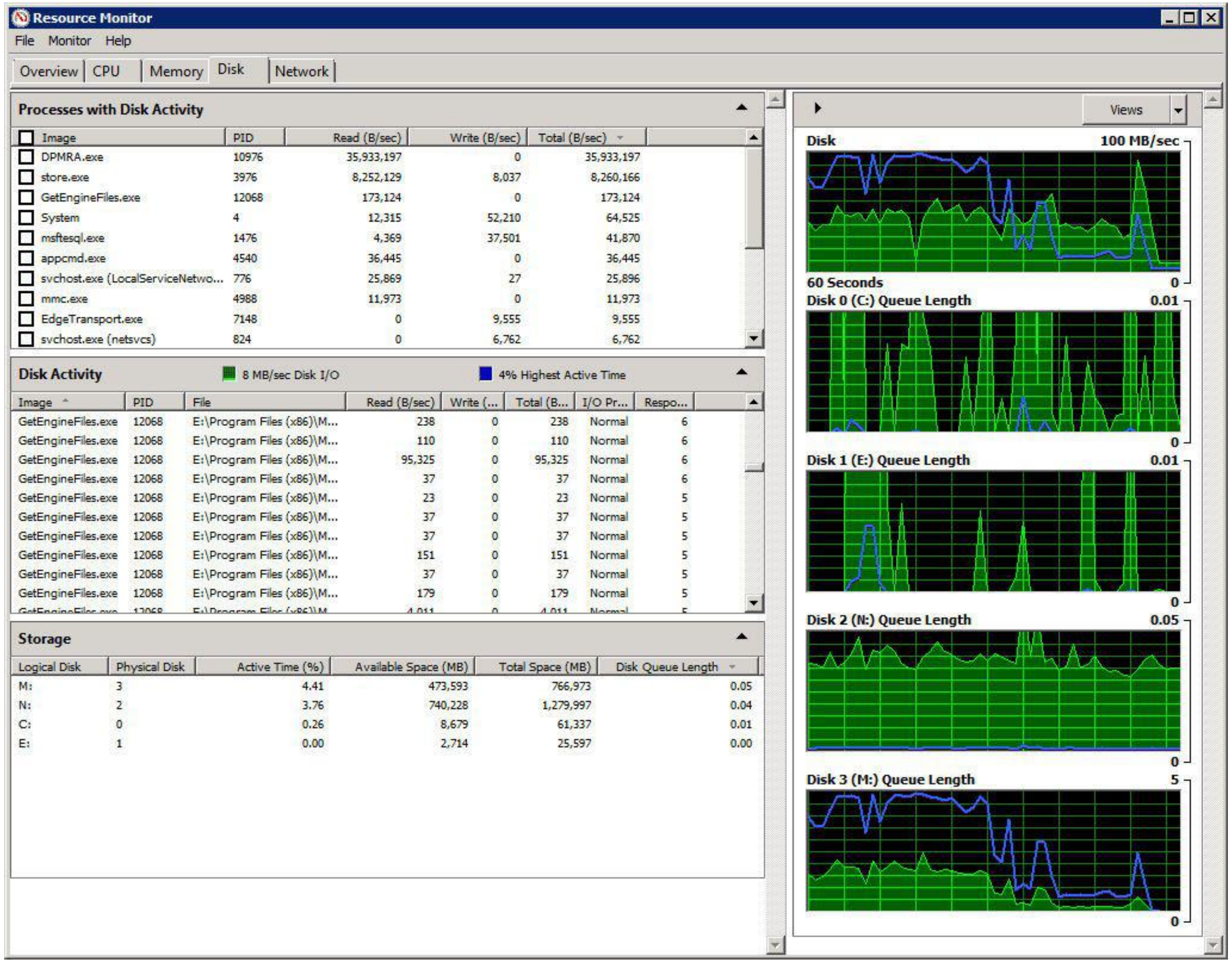
Image	PID	File	Read (B/sec)	Write (...	Total (B...	I/O Pr...	Respo...
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	238	0	238	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	110	0	110	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	95,325	0	95,325	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	23	0	23	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	151	0	151	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	179	0	179	Normal	5

Storage

Logical Disk	Physical Disk	Active Time (%)	Available Space (MB)	Total Space (MB)	Disk Queue Length
M:	3	4.41	473,593	766,973	0.05
N:	2	3.76	740,228	1,279,997	0.04
C:	0	0.26	8,679	61,337	0.01
E:	1	0.00	2,714	25,597	0.00

Data volumes

- Failure to meet SLA
- Long user wait times


Resource Monitor
File Monitor Help
Overview CPU Memory Disk Network
Views
Disk 100 MB/sec
60 Seconds 0
Disk 0 (C:) Queue Length 0.01
0
Disk 1 (E:) Queue Length 0.01
0
Disk 2 (N:) Queue Length 0.05
0
Disk 3 (M:) Queue Length 5
0

Processes with Disk Activity

□ Image	PID	Read (B/sec)	Write (B/sec)	Total (B/sec)
□ DPMRA.exe	10976	35,933,197	0	35,933,197
□ store.exe	3976	8,252,129	8,037	8,260,166
□ GetEngineFiles.exe	12068	173,124	0	173,124
□ System	4	12,315	52,210	64,525
□ msftesql.exe	1476	4,369	37,501	41,870
□ appcmd.exe	4540	36,445	0	36,445
□ svchost.exe (LocalServiceNetwo...	776	25,869	27	25,896
□ mmc.exe	4988	11,973	0	11,973
□ EdgeTransport.exe	7148	0	9,555	9,555
□ svchost.exe (netsvcs)	824	0	6,762	6,762

Disk Activity 🟩 8 MB/sec Disk I/O 🟦 4% Highest Active Time

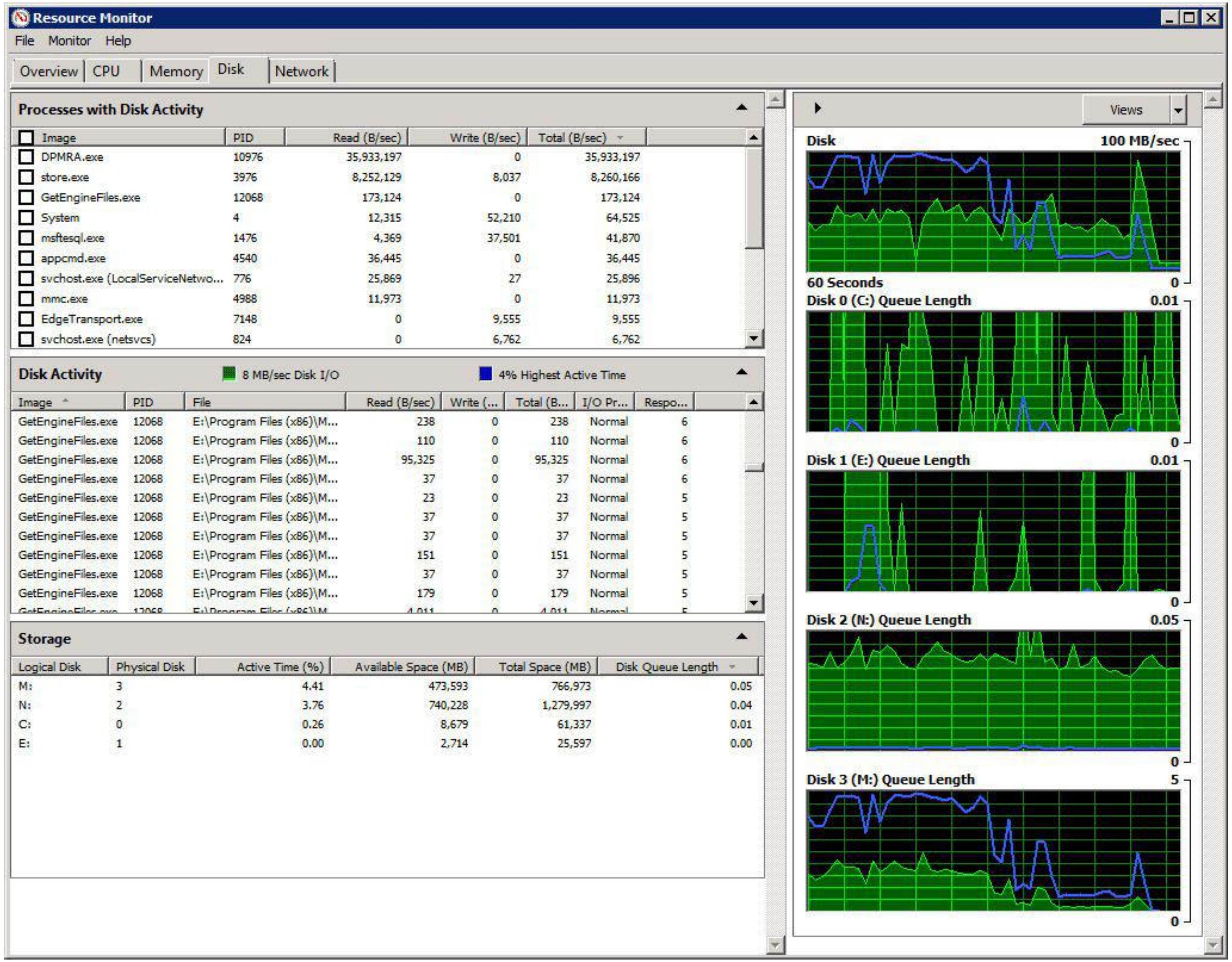
Image	PID	File	Read (B/sec)	Write (...	Total (B...	I/O Pr...	Respo...
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	238	0	238	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	110	0	110	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	95,325	0	95,325	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	23	0	23	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	151	0	151	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	179	0	179	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	4,011	0	4,011	Normal	5

Storage

Logical Disk	Physical Disk	Active Time (%)	Available Space (MB)	Total Space (MB)	Disk Queue Length
M:	3	4.41	473,593	766,973	0.05
N:	2	3.76	740,228	1,279,997	0.04
C:	0	0.26	8,679	61,337	0.01
E:	1	0.00	2,714	25,597	0.00

Data volumes

- Failure to meet SLA
- Long user wait times
- Reports generate high strain → slow


Resource Monitor
File Monitor Help
Overview CPU Memory Disk Network
Views
Disk 100 MB/sec
60 Seconds 0
Disk 0 (C:) Queue Length 0.01
0
Disk 1 (E:) Queue Length 0.01
0
Disk 2 (N:) Queue Length 0.05
0
Disk 3 (M:) Queue Length 5
0

Processes with Disk Activity

□ Image	PID	Read (B/sec)	Write (B/sec)	Total (B/sec)
□ DPMRA.exe	10976	35,933,197	0	35,933,197
□ store.exe	3976	8,252,129	8,037	8,260,166
□ GetEngineFiles.exe	12068	173,124	0	173,124
□ System	4	12,315	52,210	64,525
□ msftesql.exe	1476	4,369	37,501	41,870
□ appcmd.exe	4540	36,445	0	36,445
□ svchost.exe (LocalServiceNetwo...	776	25,869	27	25,896
□ mmc.exe	4988	11,973	0	11,973
□ EdgeTransport.exe	7148	0	9,555	9,555
□ svchost.exe (netsvcs)	824	0	6,762	6,762

Disk Activity 🟩 8 MB/sec Disk I/O 🟦 4% Highest Active Time

Image	PID	File	Read (B/sec)	Write (...	Total (B...	I/O Pr...	Respo...
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	238	0	238	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	110	0	110	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	95,325	0	95,325	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	6
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	23	0	23	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	151	0	151	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	37	0	37	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	179	0	179	Normal	5
GetEngineFiles.exe	12068	E:\Program Files (x86)\M...	4,011	0	4,011	Normal	5

Storage

Logical Disk	Physical Disk	Active Time (%)	Available Space (MB)	Total Space (MB)	Disk Queue Length
M:	3	4.41	473,593	766,973	0.05
N:	2	3.76	740,228	1,279,997	0.04
C:	0	0.26	8,679	61,337	0.01
E:	1	0.00	2,714	25,597	0.00

Reproducibility

- Rerun parts of your data pipeline without thinking?

Reproducibility

- Rerun parts of your data pipeline without thinking?
- Can you regenerate your entire warehouse (in principle)?

Reproducibility

- Rerun parts of your data pipeline without thinking?
- Can you regenerate your entire warehouse (in principle)?
- → Easy to solve bugs

Limitations of ETL tooling

- Focused on a specific database

Limitations of ETL tooling

- Focused on a specific database
- Not scalable

Limitations of ETL tooling

- Focused on a specific database
- Not scalable
- Difficult to synchronize with other scheduled pipelines

Limitations of ETL tooling

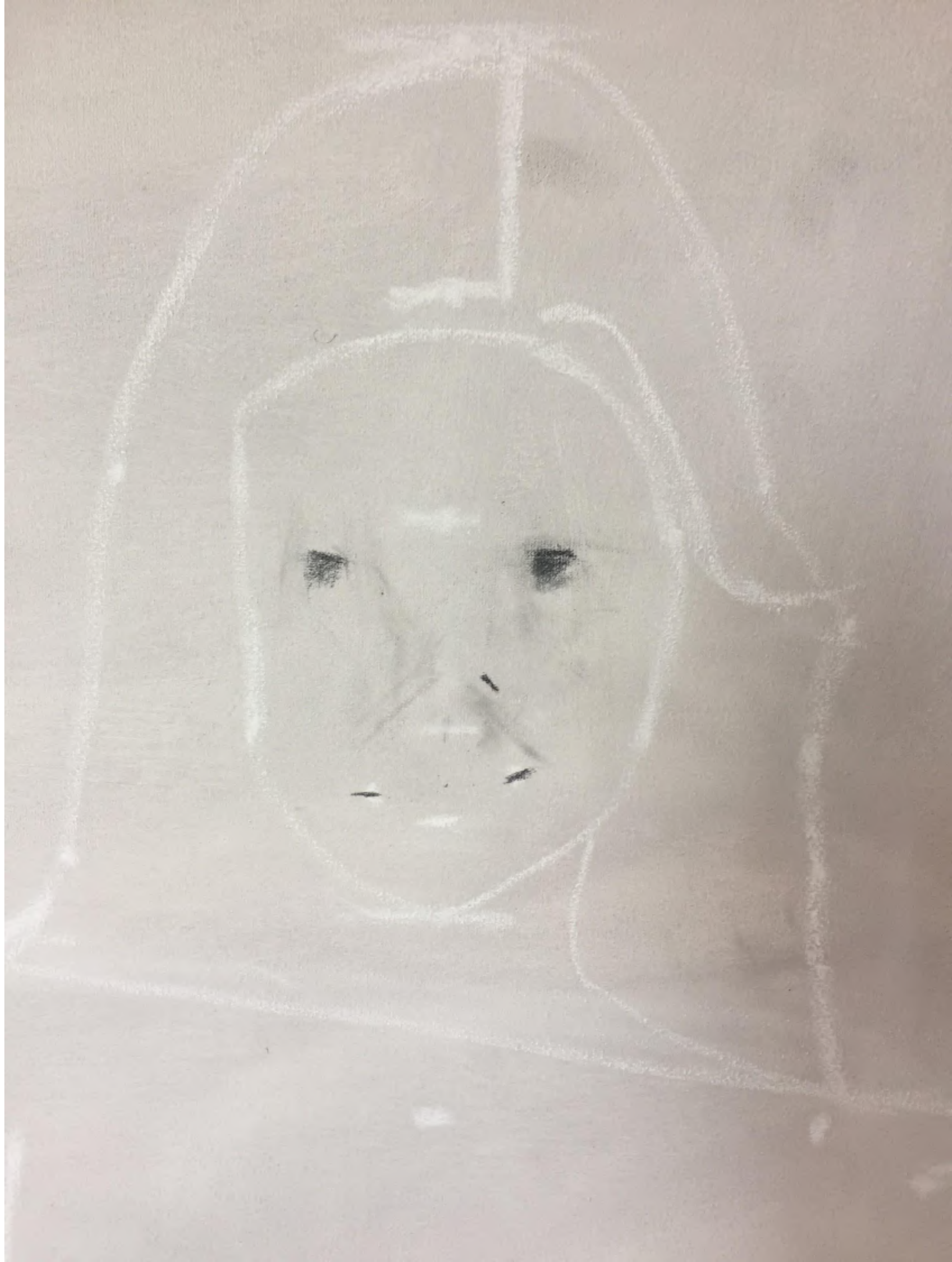
- Focused on a specific database
- Not scalable
- Difficult to synchronize with other scheduled pipelines
- Not built from a functional philosophy

Limitations of ETL tooling

- Focused on a specific database
- Not scalable
- Difficult to synchronize with other scheduled pipelines
- Not built from a functional philosophy
- Not extendable as a platform

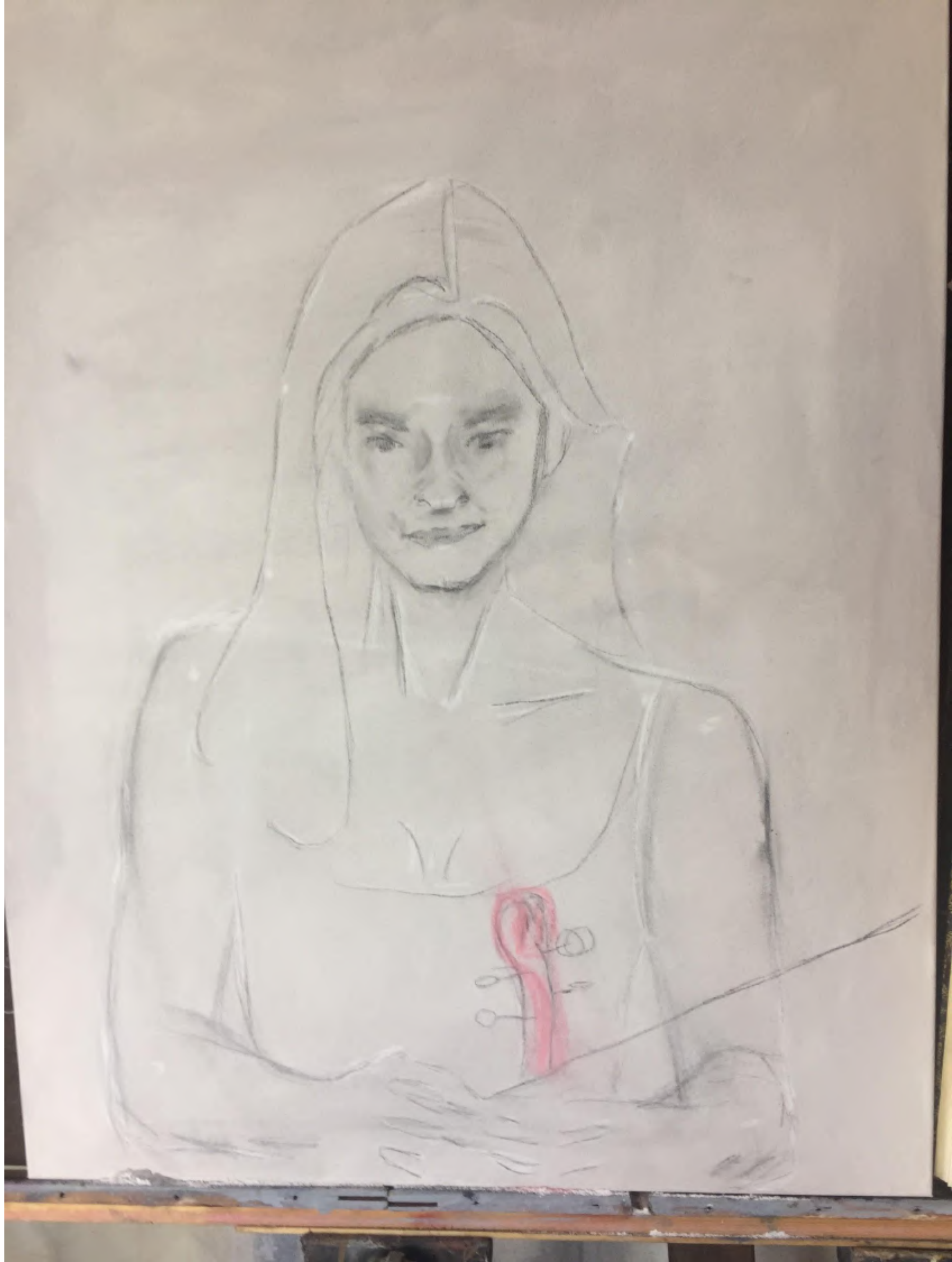
Engineering is a methodological process of stages requiring:

- engineering skills
- knowing what to do when
- and what NOT to do when

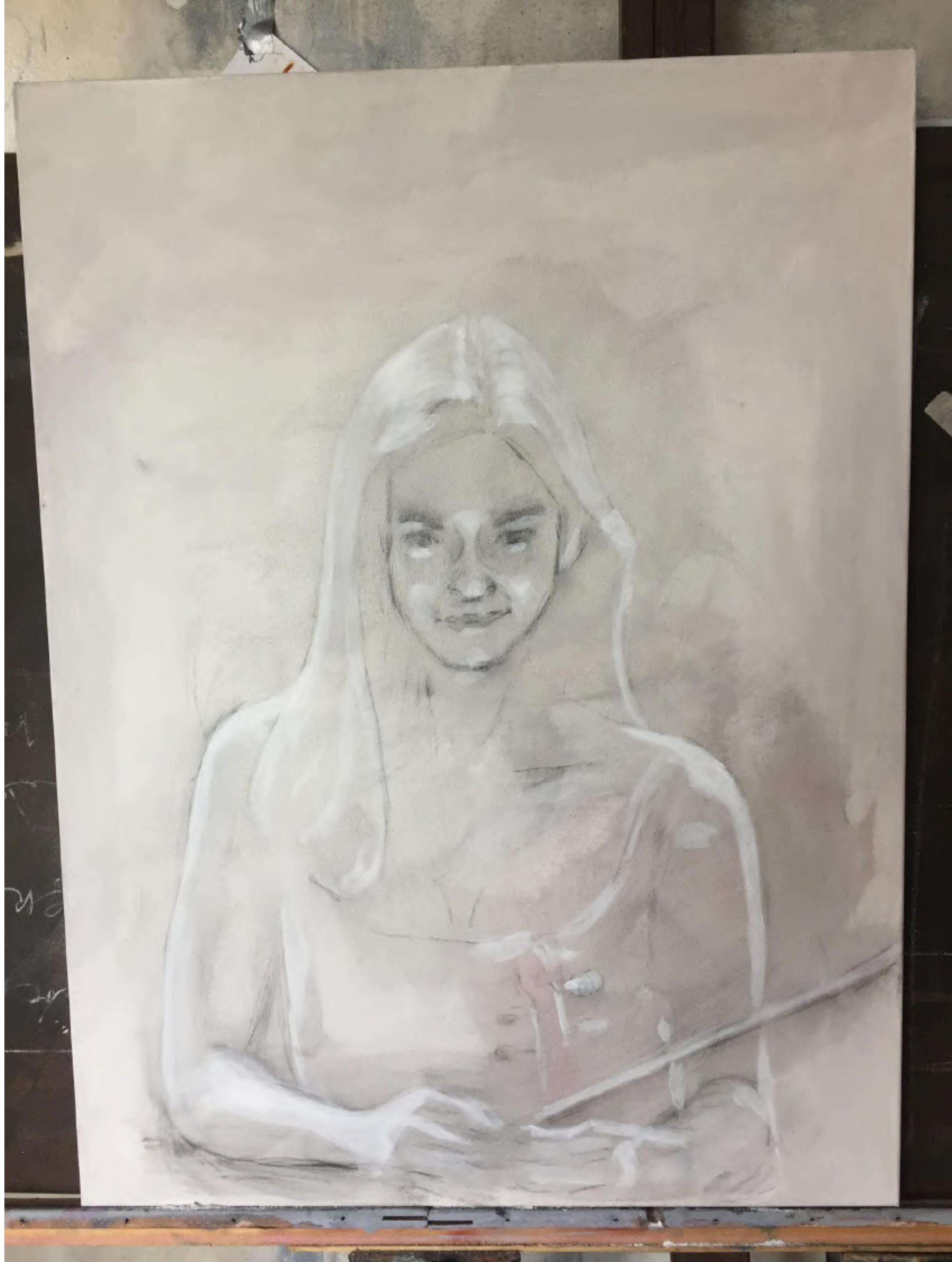


The concept

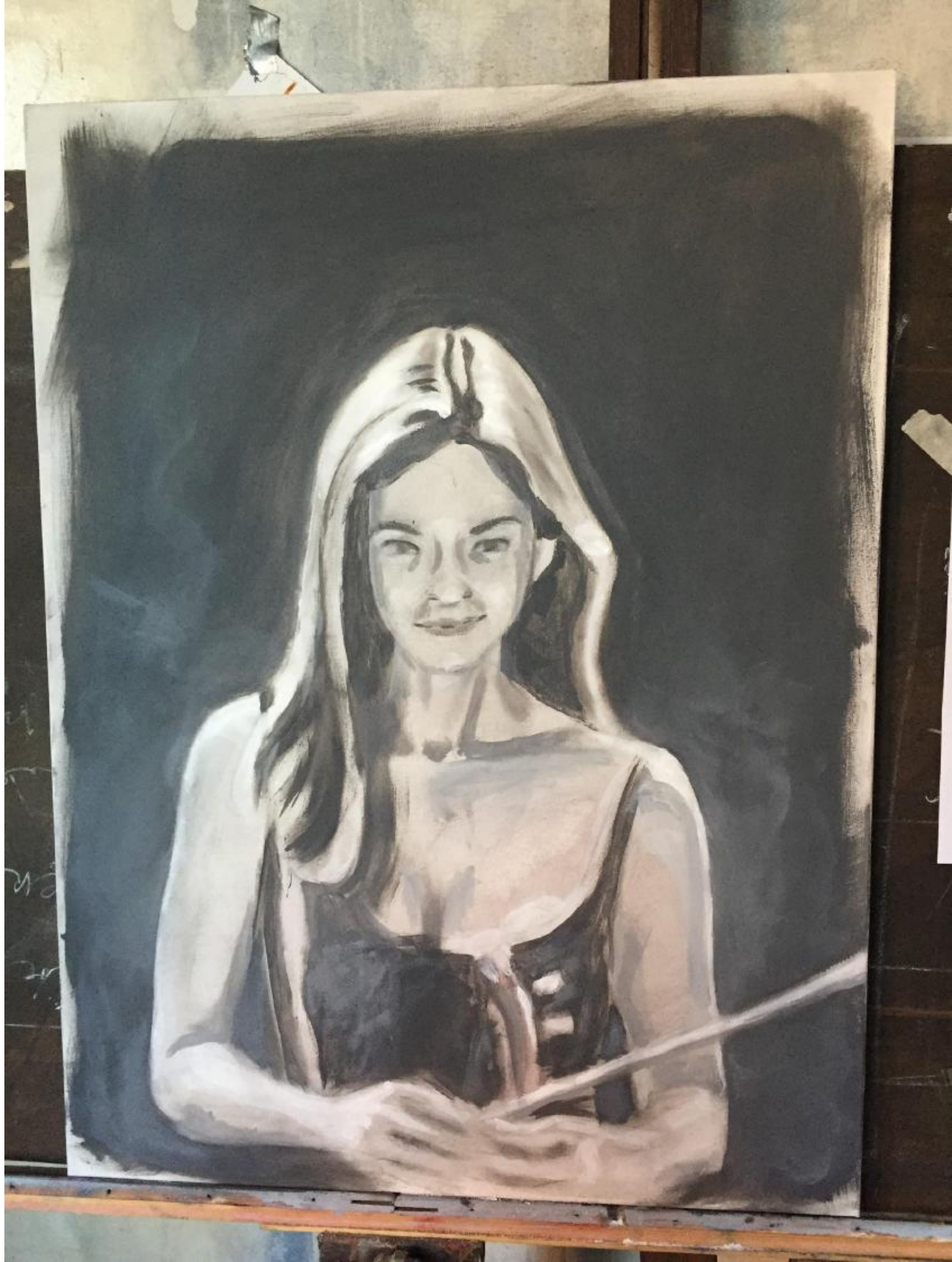
Contextualization



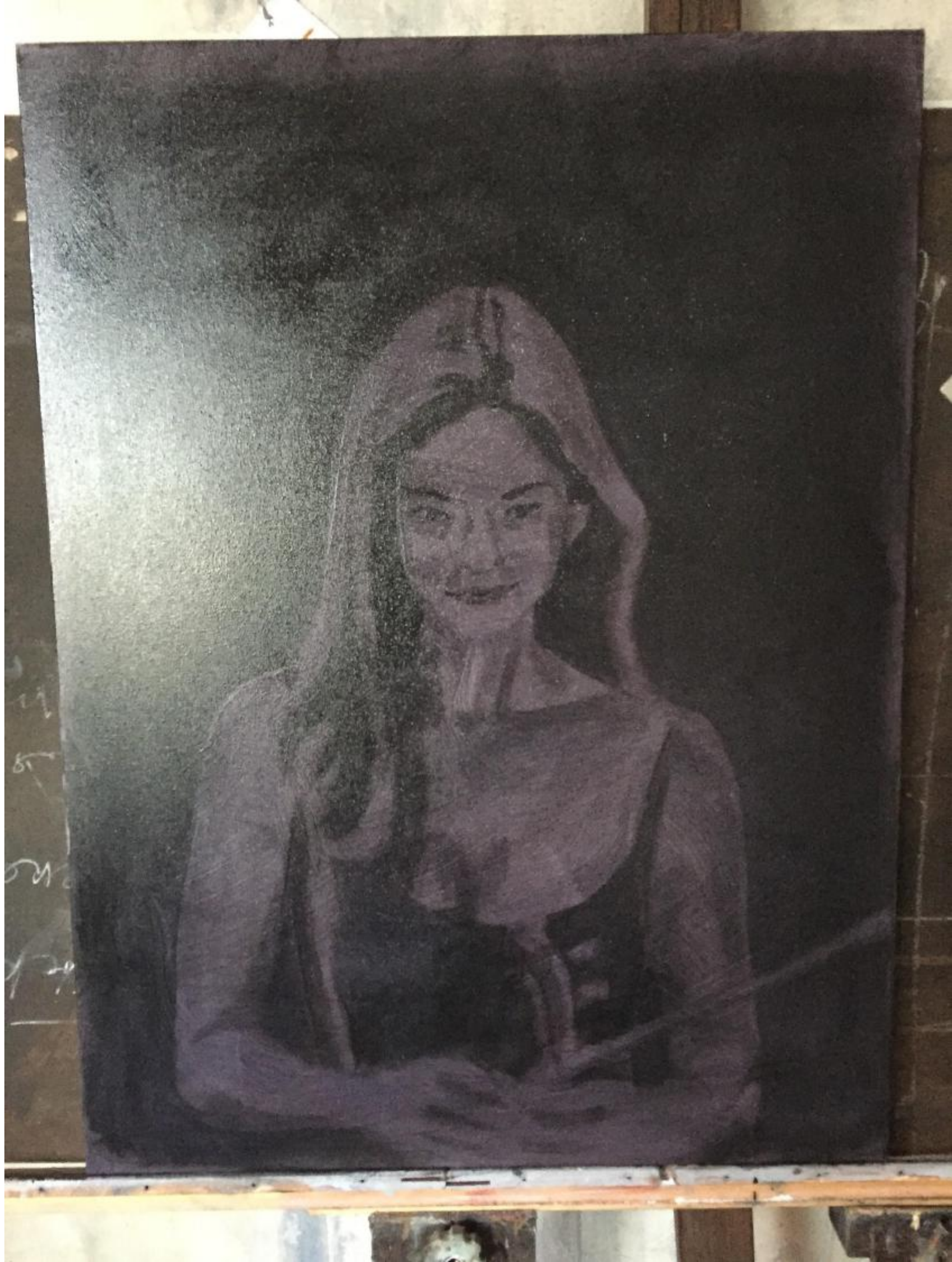
The underlayer



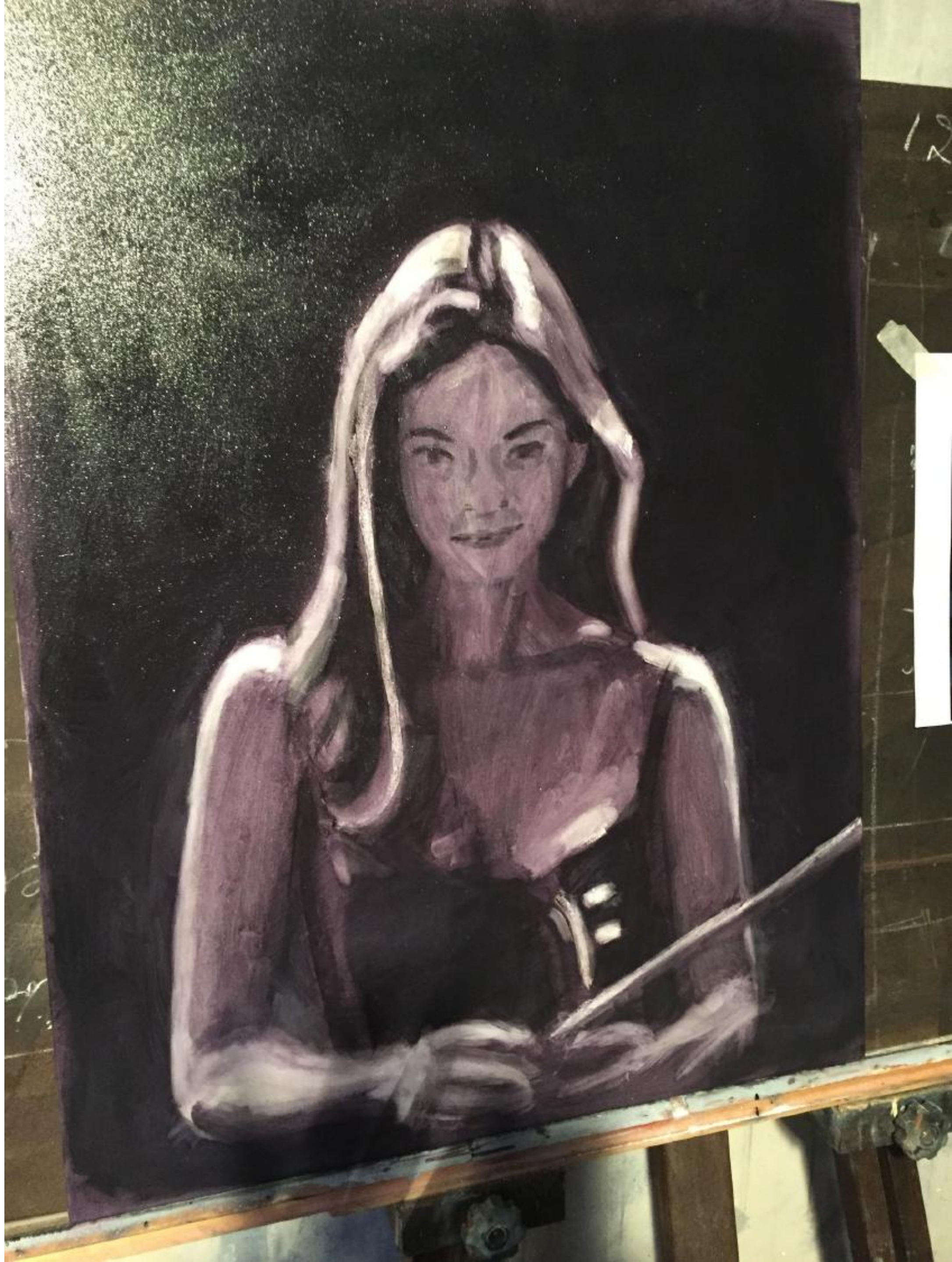
Finished underlayer



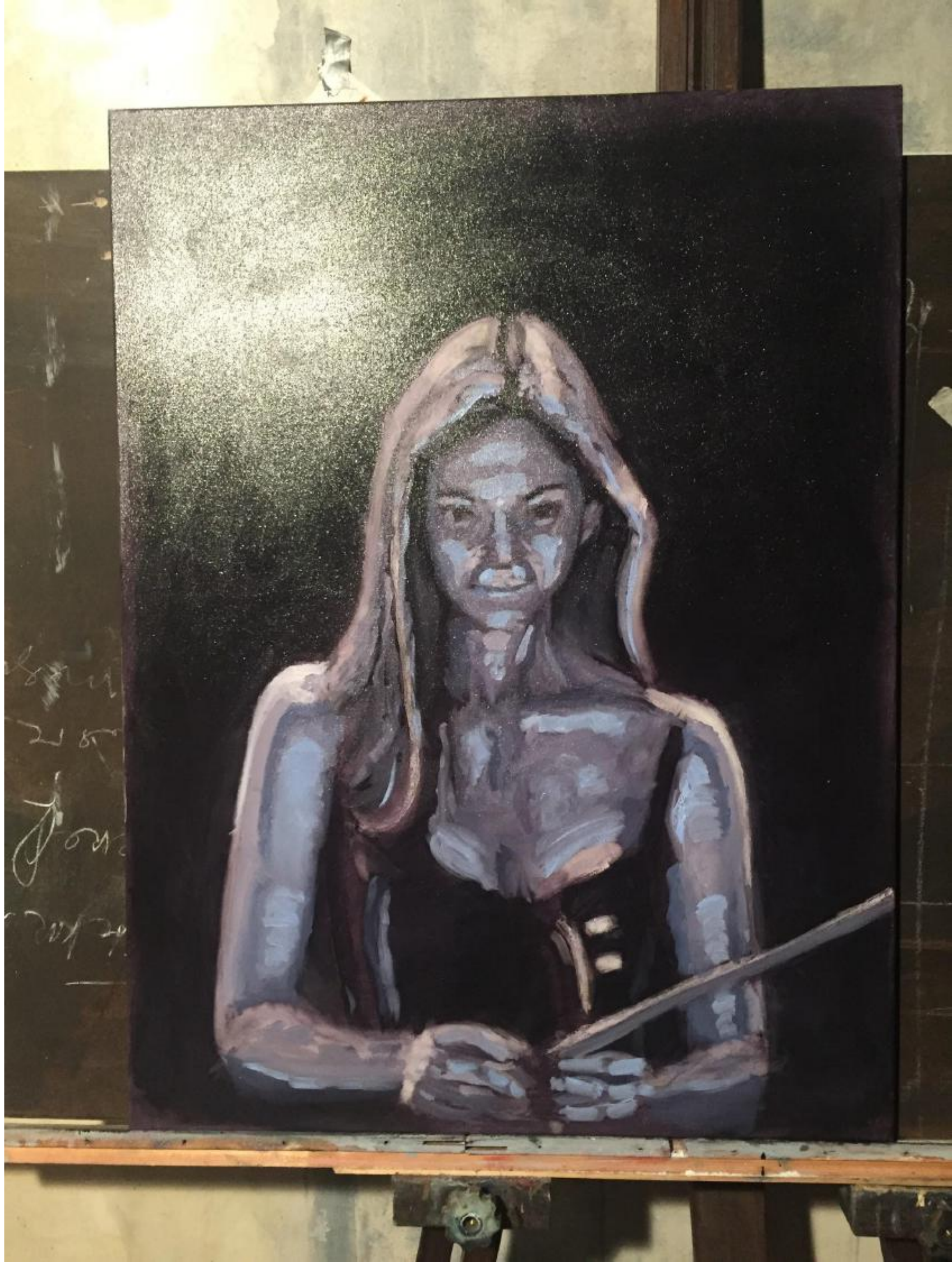
Painting the final layer

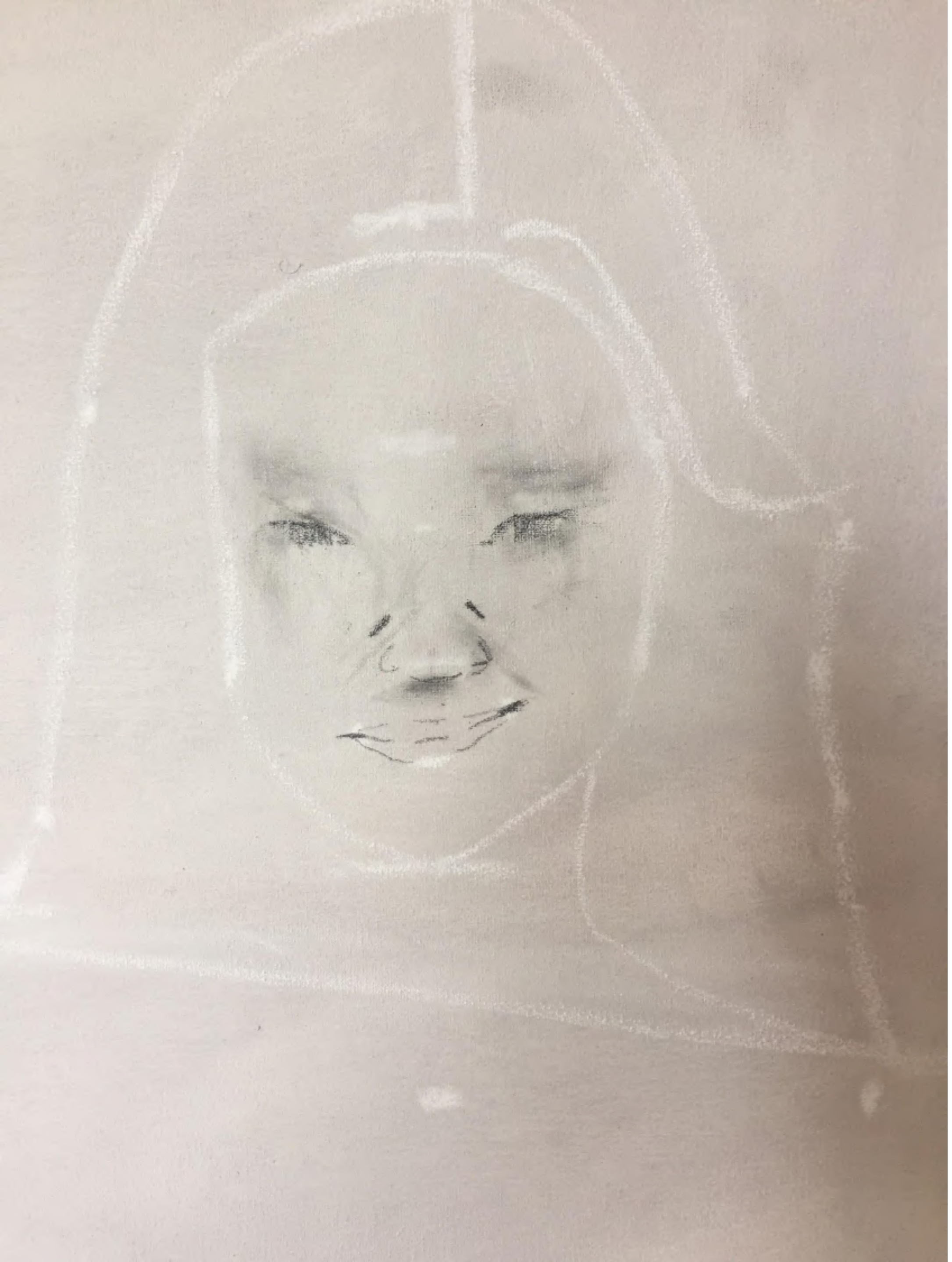


Reductive as well as additive



Add color

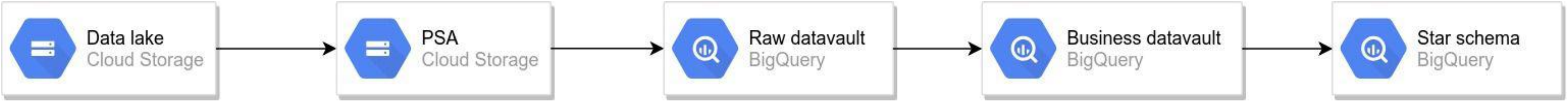




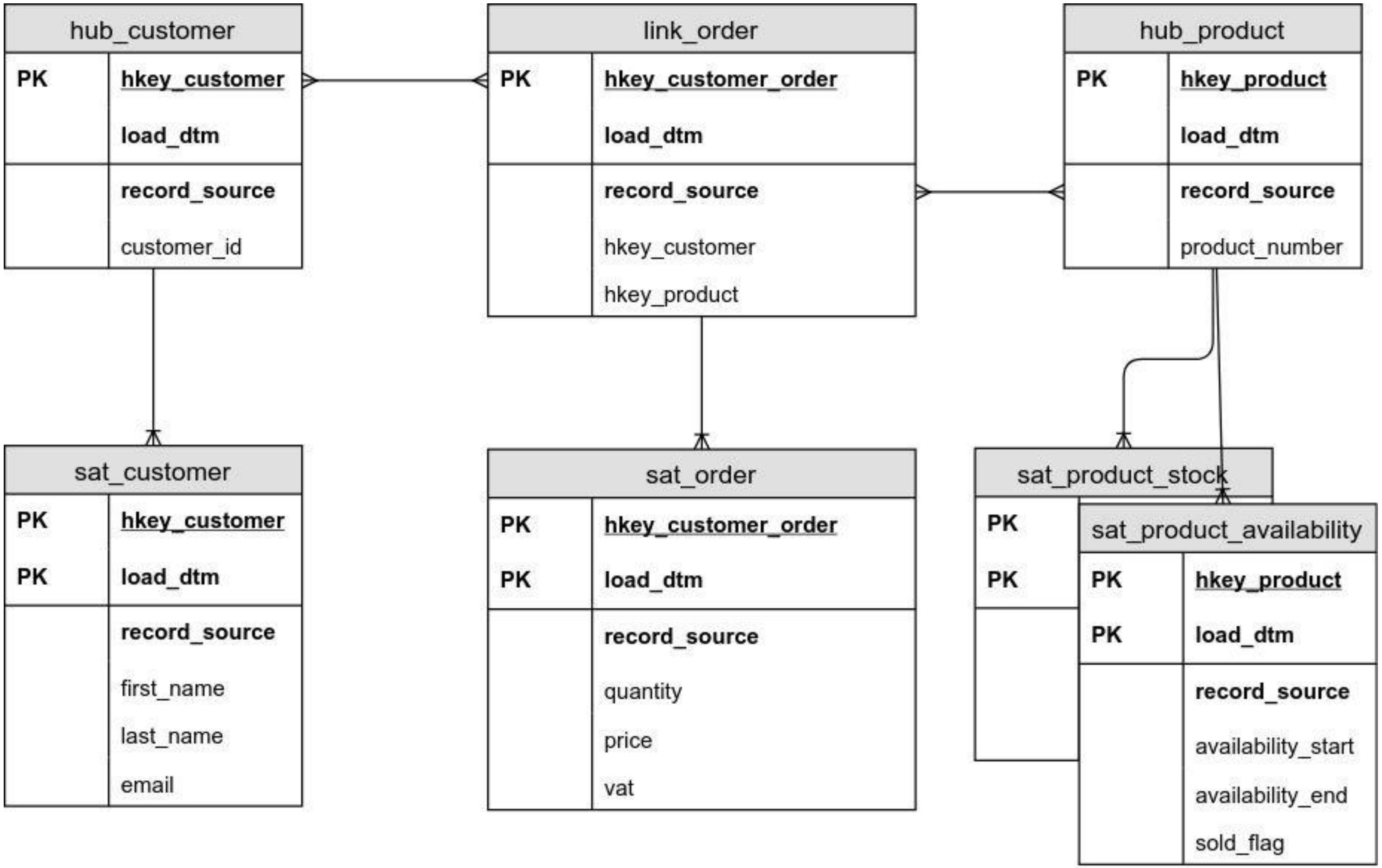
Agile Data Architecture

- Choose good ETL tools with generalized, reusable components
- Apply design decisions at the right stage, not too early
- Build ‘functional’ data pipelines
- Make pipelines reproducible → immutable partitions
- Decouple the source schema from your warehouse schema
- Don’t (immediately) resist more persistent copies of the data in pipeline

The overall data pipeline



What is a (raw) data vault 2.0?



hub_customer	
PK	**hkey_customer**
	load_dtm
	record_source
	customer_id

link_order	
PK	**hkey_customer_order**
	load_dtm
	record_source
	hkey_customer
	hkey_product

hub_product	
PK	**hkey_product**
	load_dtm
	record_source
	product_number

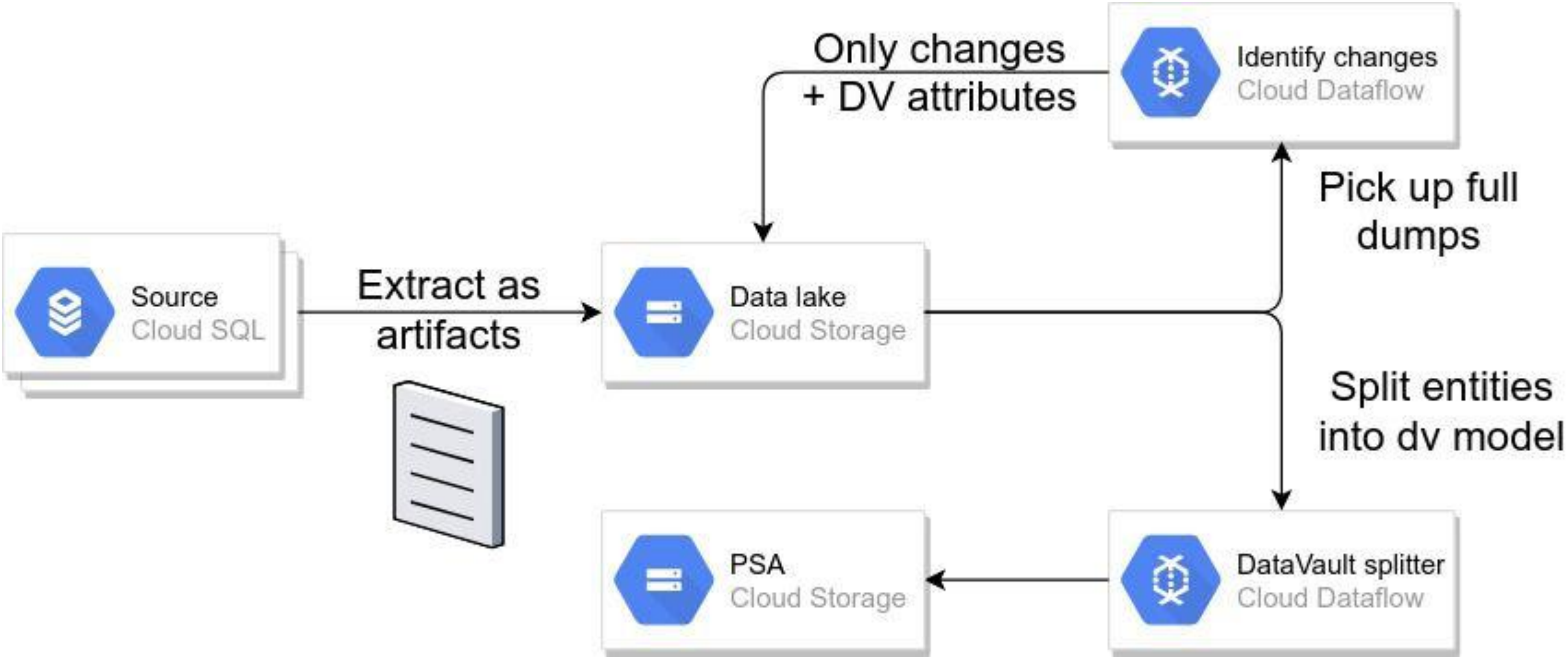
sat_customer	
PK	**hkey_customer**
PK	**load_dtm**
	record_source
	first_name
	last_name
	email

sat_order	
PK	**hkey_customer_order**
PK	**load_dtm**
	record_source
	quantity
	price
	vat

sat_product_stock	
PK	
PK	

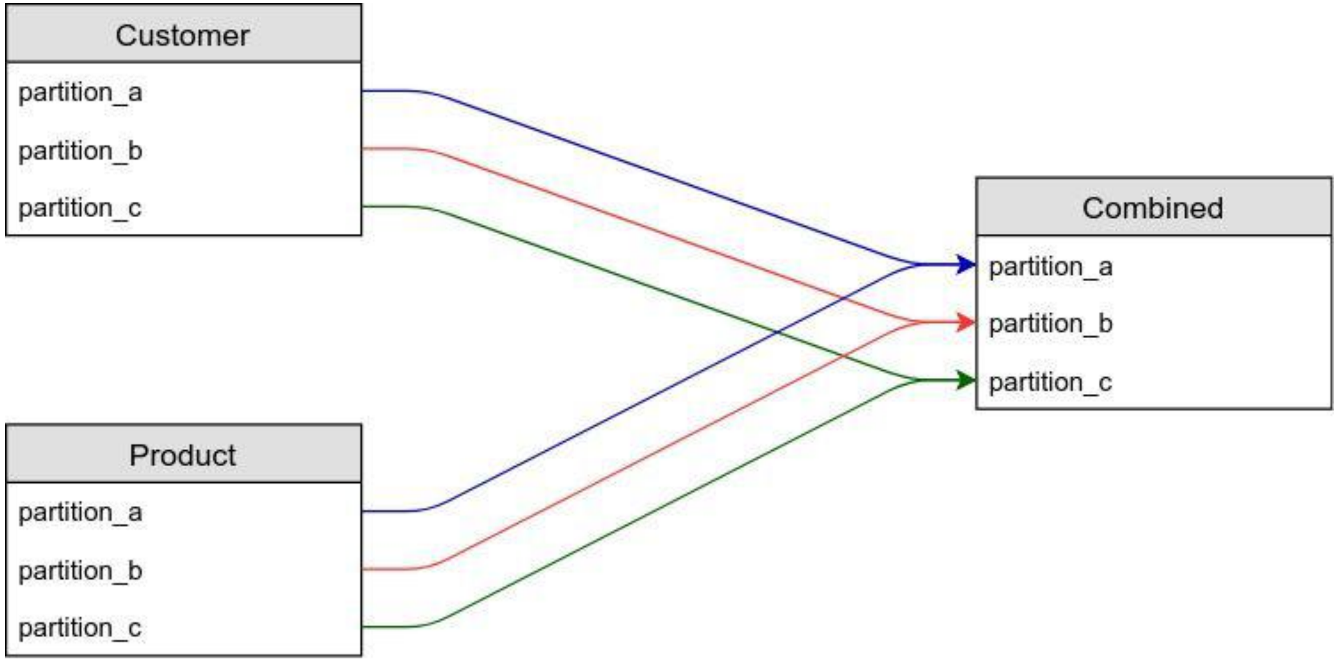
sat_product_availability	
PK	**hkey_product**
PK	**load_dtm**
	record_source
	availability_start
	availability_end
	sold_flag

Step 1: Break data into reloadable partitions

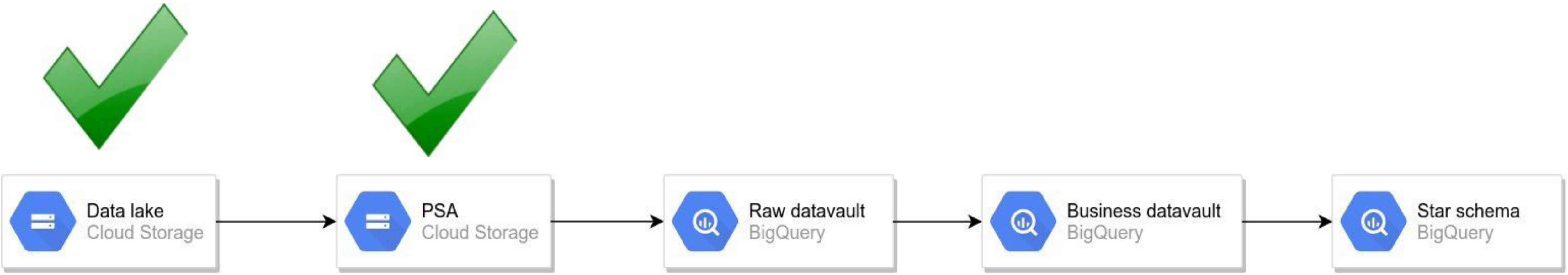


Step 1: Break data into reloadable partitions

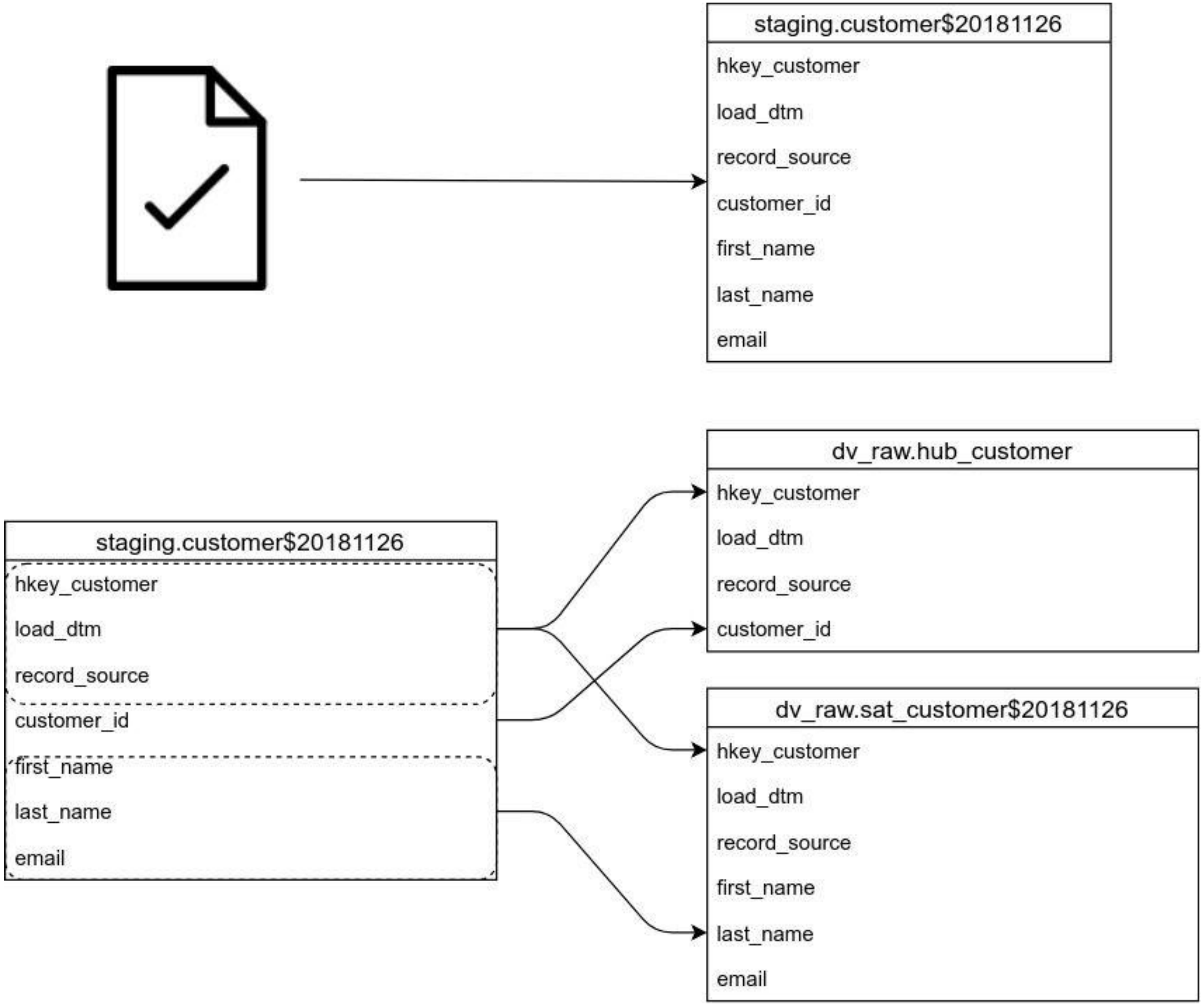
gs://datalake/datavault/psa/crm/customer/2018/11/26/data.avro	partition_a (customer)
gs://datalake/datavault/psa/crm/customer/2018/11/27/data.avro	partition_b (customer)
gs://datalake/datavault/psa/crm/customer/2018/11/28/data.avro	partition_c (customer)
gs://datalake/datavault/psa/crm/product/2018/11/26/data.avro	partition_a (product)
gs://datalake/datavault/psa/crm/product/2018/11/27/data.avro	partition_b (product)
gs://datalake/datavault/psa/crm/product/2018/11/28/data.avro	partition_c (product)



The overall architecture



Step 2: Load raw datavault



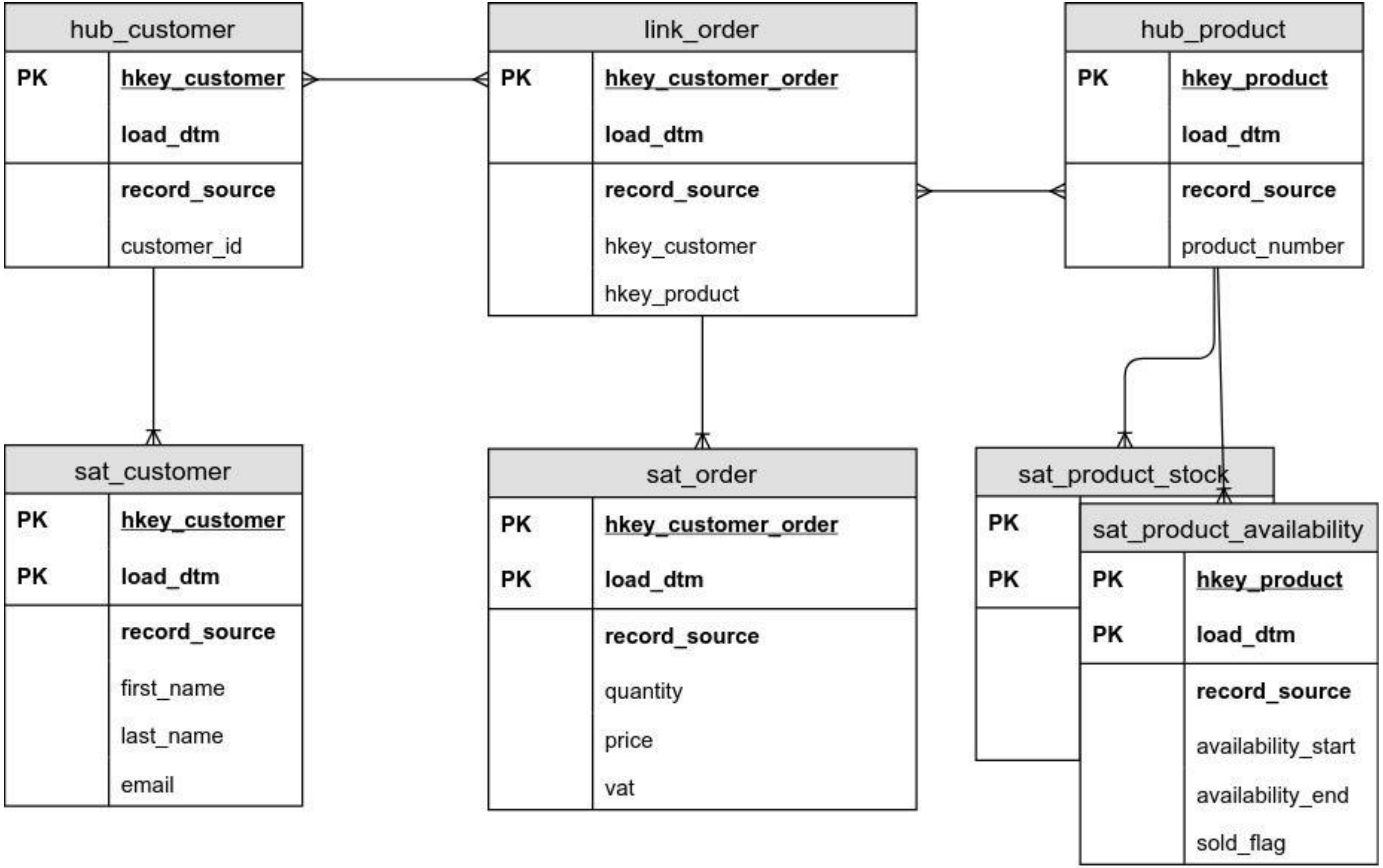
staging.customer$20181126
- hkey_customer
- load_dtm
- record_source
- customer_id
- first_name
- last_name
- email

staging.customer$20181126
- hkey_customer
- load_dtm
- record_source
- customer_id
- first_name
- last_name
- email

dv_raw.hub_customer
- hkey_customer
- load_dtm
- record_source
- customer_id

dv_raw.sat_customer$20181126
- hkey_customer
- load_dtm
- record_source
- first_name
- last_name
- email

Do this for all entities



hub_customer	
PK	**hkey_customer**
	load_dtm
	record_source
	customer_id

link_order	
PK	**hkey_customer_order**
	load_dtm
	record_source
	hkey_customer
	hkey_product

hub_product	
PK	**hkey_product**
	load_dtm
	record_source
	product_number

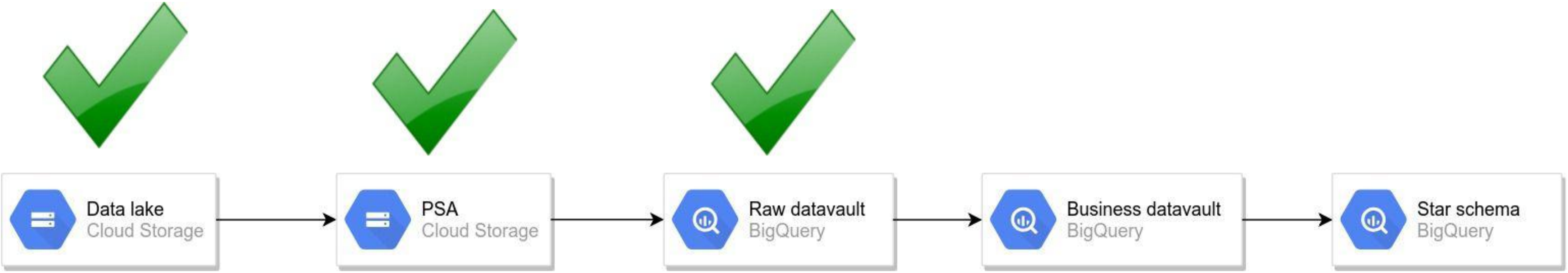
sat_customer	
PK	**hkey_customer**
PK	**load_dtm**
	record_source
	first_name
	last_name
	email

sat_order	
PK	**hkey_customer_order**
PK	**load_dtm**
	record_source
	quantity
	price
	vat

sat_product_stock	
PK	
PK	

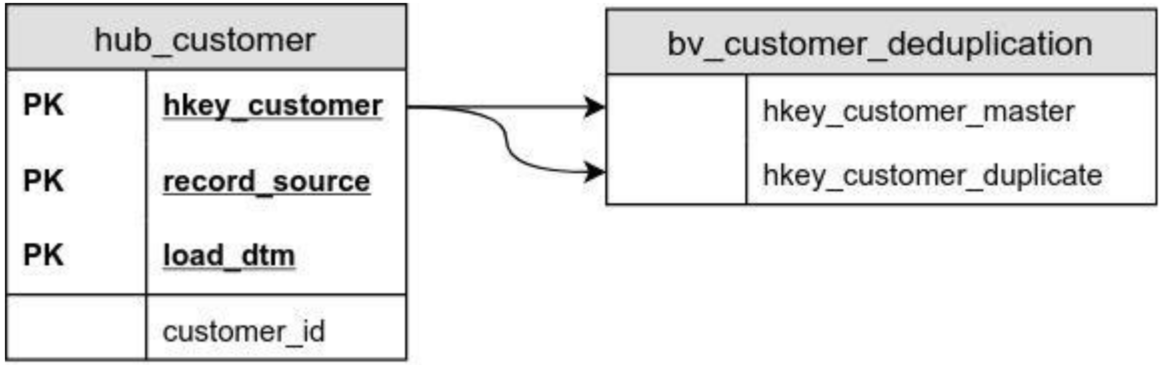
sat_product_availability	
PK	**hkey_product**
PK	**load_dtm**
	record_source
	availability_start
	availability_end
	sold_flag

The overall architecture



The business vault (data interpretation)

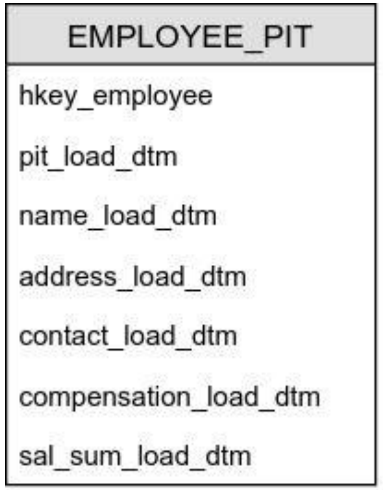
Customer deduplication:



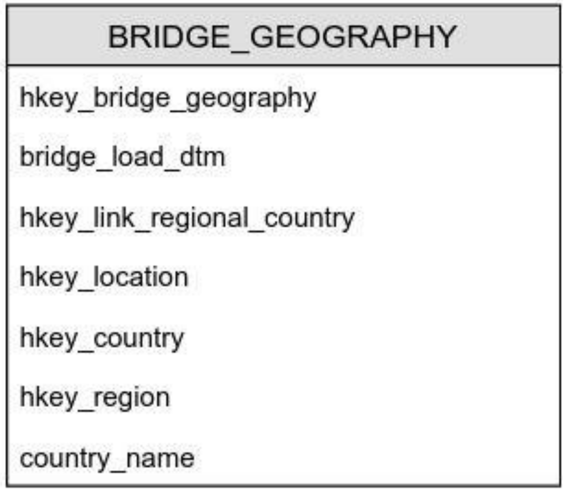
hub_customer	
PK	**hkey_customer**
PK	**record_source**
PK	**load_dtm**
	customer_id

bv_customer_deduplication	
	hkey_customer_master
	hkey_customer_duplicate

Bridge & PIT tables:



EMPLOYEE_PIT
hkey_employee
pit_load_dtm
name_load_dtm
address_load_dtm
contact_load_dtm
compensation_load_dtm
sal_sum_load_dtm

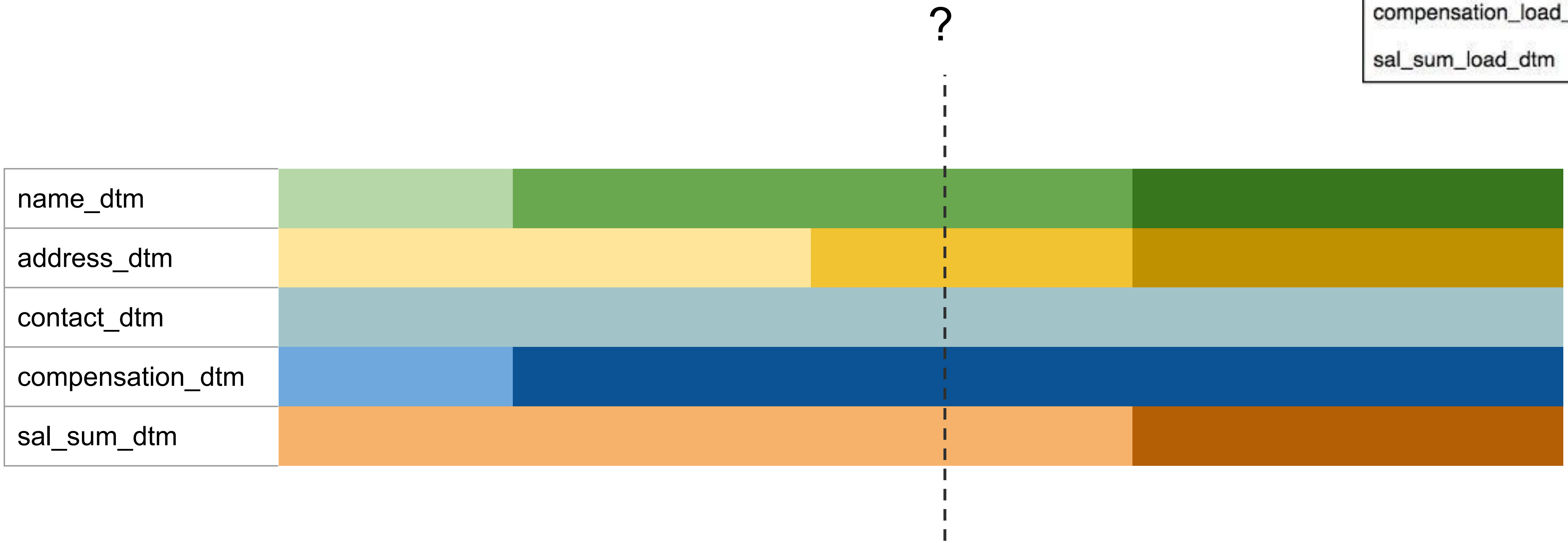


BRIDGE_GEOGRAPHY
hkey_bridge_geography
bridge_load_dtm
hkey_link_regional_country
hkey_location
hkey_country
hkey_region
country_name

Calculations, data cleaning, data quality, soft business rules

The PIT table (data interpretation)

EMPLOYEE_PIT
hkey_employee
pit_load_dtm
name_load_dtm
address_load_dtm
contact_load_dtm
compensation_load_dtm
sal_sum_load_dtm



The Bridge table (data interpretation)

BRIDGE_GEOGRAPHY

hkey_bridge_geography

bridge_load_dtm

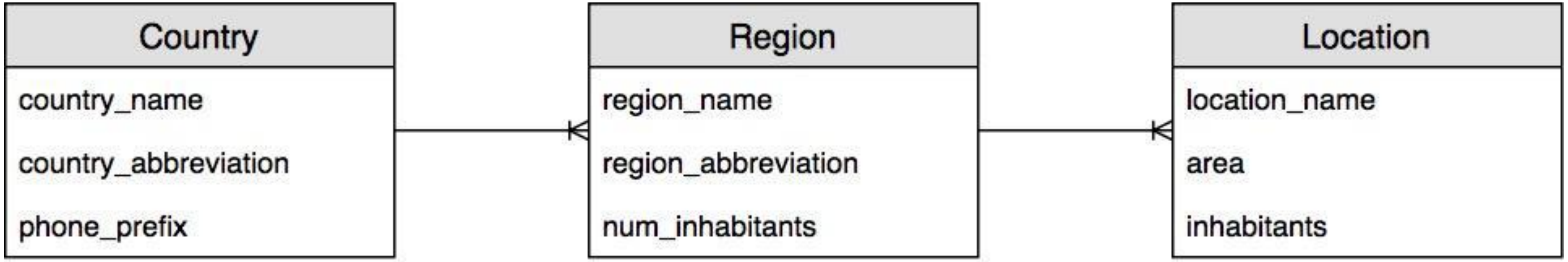
hkey_link_regional_country

hkey_location

hkey_country

hkey_region

country_name



Country

country_name

country_abbreviation

phone_prefix

Region

region_name

region_abbreviation

num_inhabitants

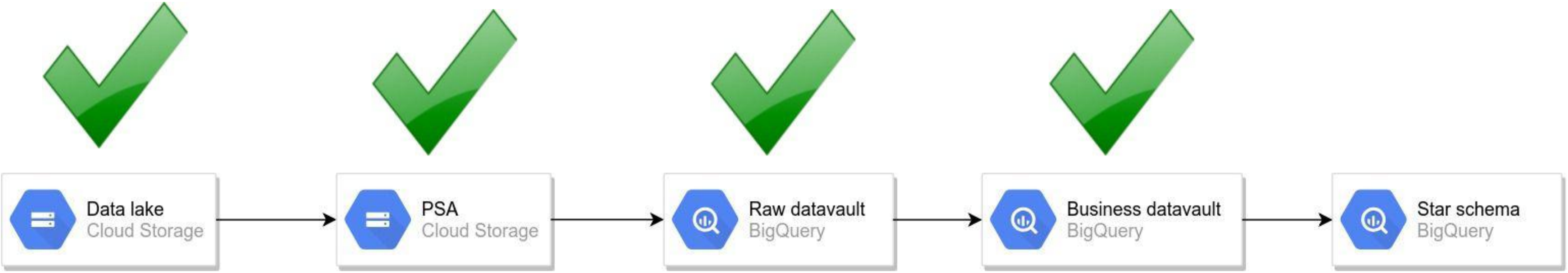
Location

location_name

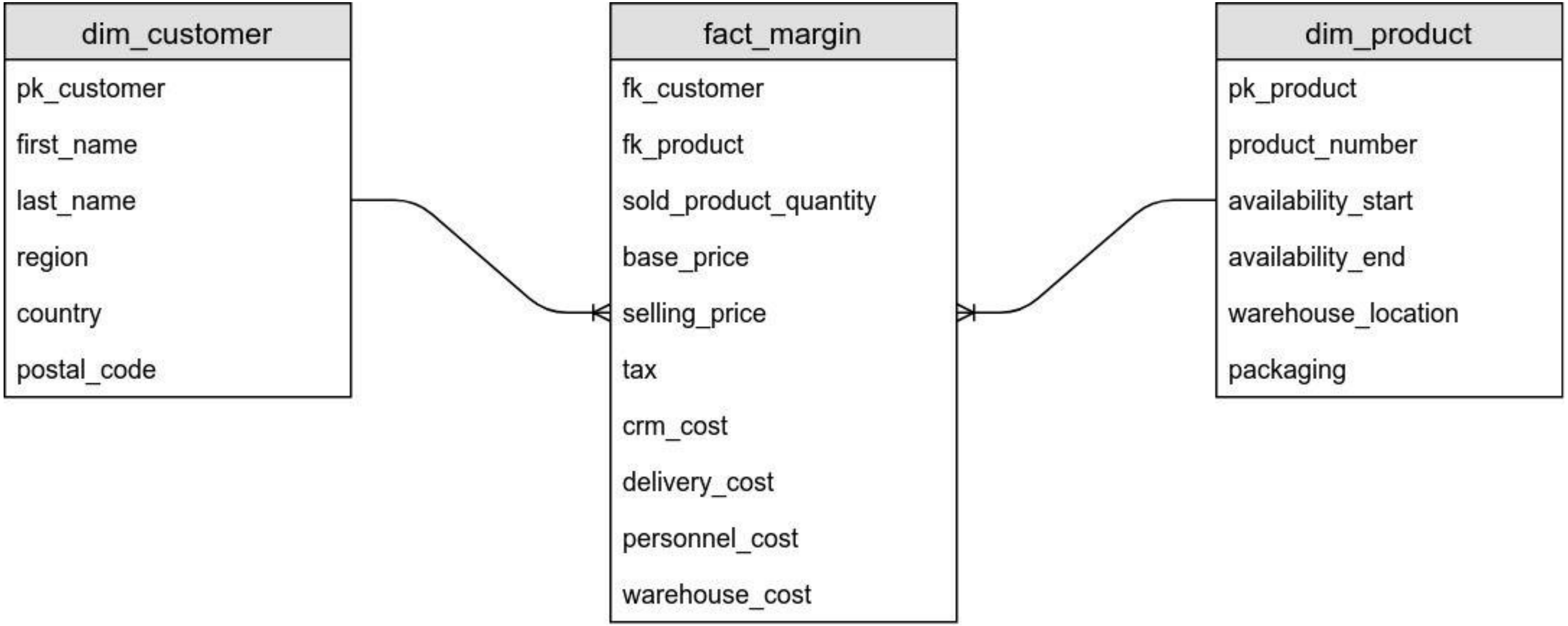
area

inhabitants

The overall architecture



The star schema (data interpretation)



dim_customer

- pk_customer
- first_name
- last_name
- region
- country
- postal_code

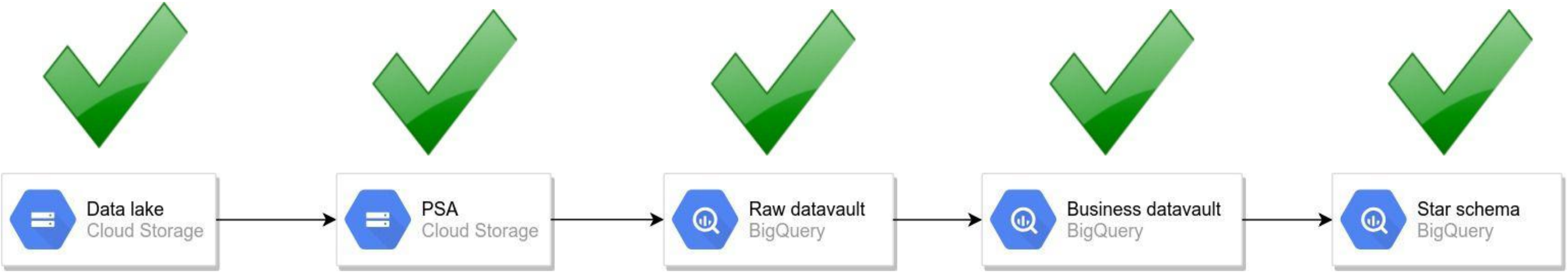
fact_margin

- fk_customer
- fk_product
- sold_product_quantity
- base_price
- selling_price
- tax
- crm_cost
- delivery_cost
- personnel_cost
- warehouse_cost

dim_product

- pk_product
- product_number
- availability_start
- availability_end
- warehouse_location
- packaging

The overall architecture



Data lake
Cloud Storage

PSA
Cloud Storage

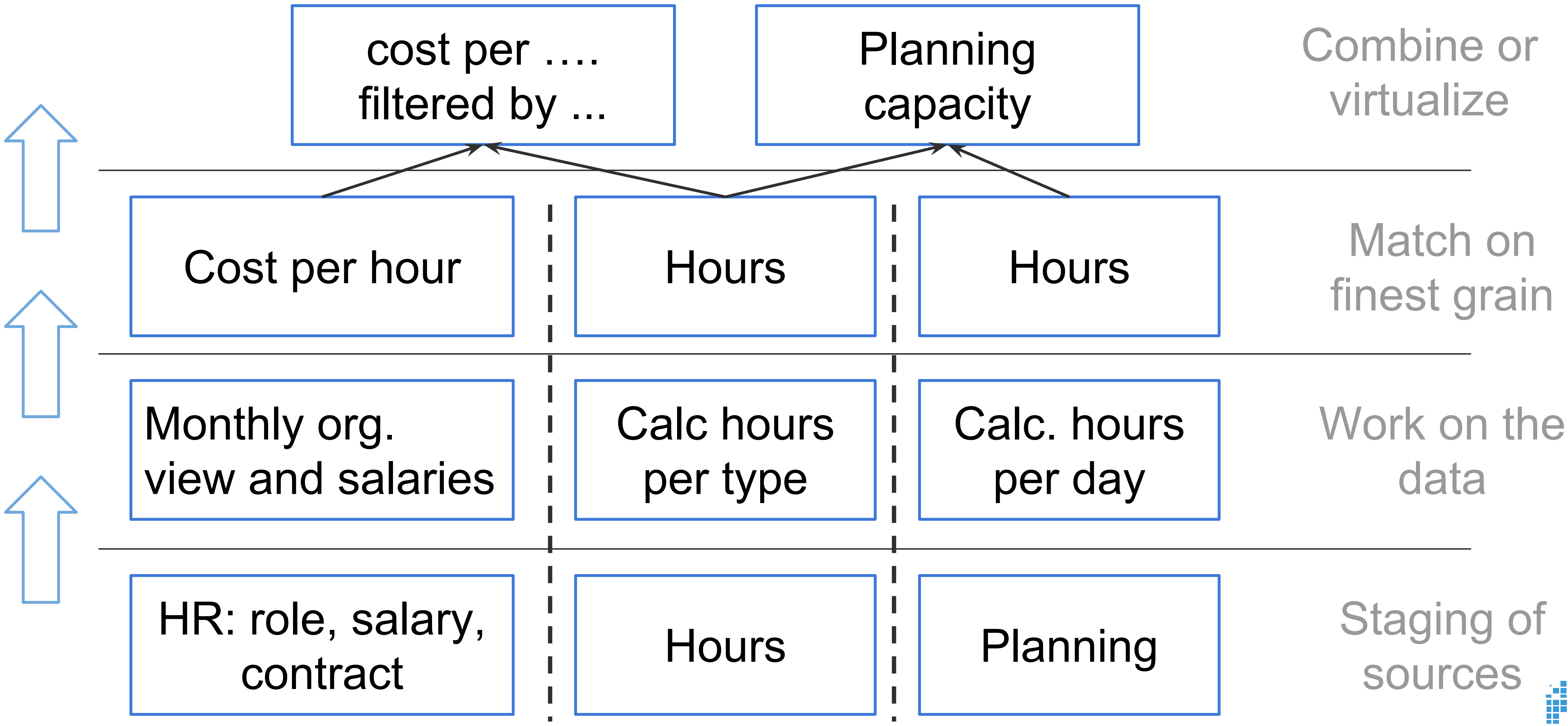
Raw datavault
BigQuery

Business datavault
BigQuery

Star schema
BigQuery

Avoid complication with “data lanes”

Merge data from different sources as late as possible



cost per ….
filtered by ...

Planning
capacity

Combine or
virtualize

Cost per hour

Hours

Hours

Match on
finest grain

Monthly org.
view and salaries

Calc hours
per type

Calc. hours
per day

Work on the
data

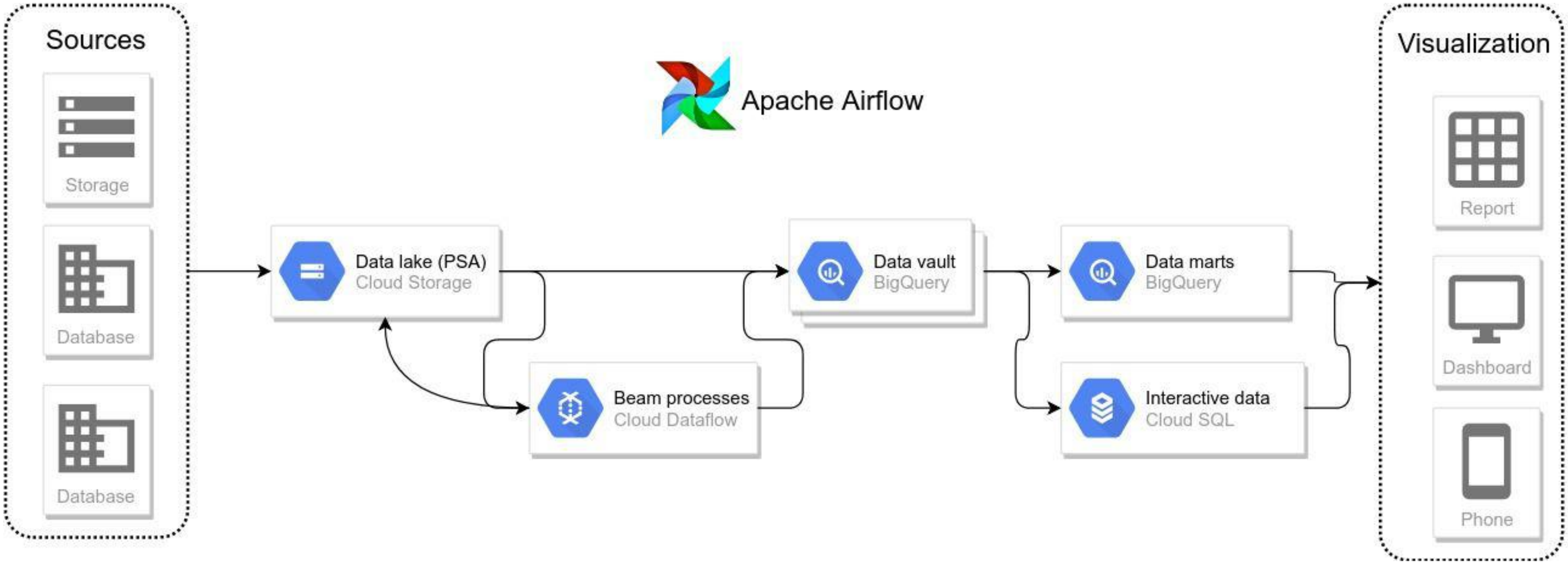
HR: role, salary,
contract

Hours

Planning

Staging of
sources

A possible architecture



Join at
slido.com: #bigdata2018

bigdata REPUBLIC

DATA SCIENCE | BIG DATA ANALYTICS | BIG DATA ARCHITECTURES

☎ +31 (0)1 - 68479294

🏠 Coltbaan 4C, Nieuwegein

🐦 @bigdatarep

🌐 www.bigdatarepublic.nl

in /company/bigdata-republic

✉ info@bigdatarepublic.nl