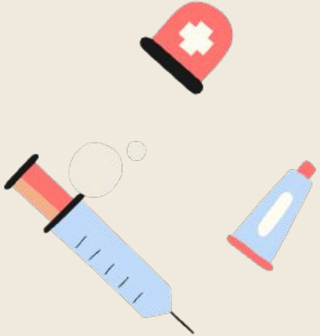
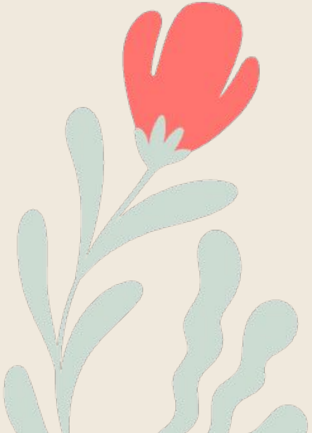


Group 7

Novartis Healthcare Data Management System

Krish Engineer, Monica Martinez, Pratik Gawli,
Sai Bhargav Tetali, Vivian Wang



Overview

1. Introduction
2. DDL and Data Models
3. Data Warehouse ETL
4. DataMarts and DataLake
5. Analytics

01 INTRODUCTION

Novartis Healthcare has a data management system storing tons of data corresponding to various transactional operations ranging from drug sales to drug promotional campaigns

In this project, we consider the following three transactional operations:

- Drug sales Data
- Inventory Data
- Drug promotional campaign Data

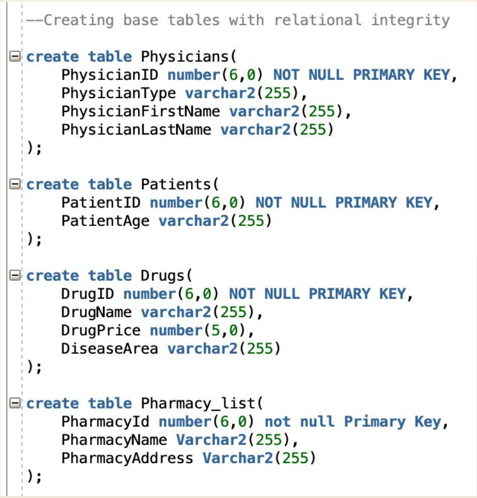
Addressing the normalizations for each of these operations, we develop a Data management system using a two tier Data warehouse to denormalize and store data to make it usable for analytical operations and drawing business Insights

02

DDL Code

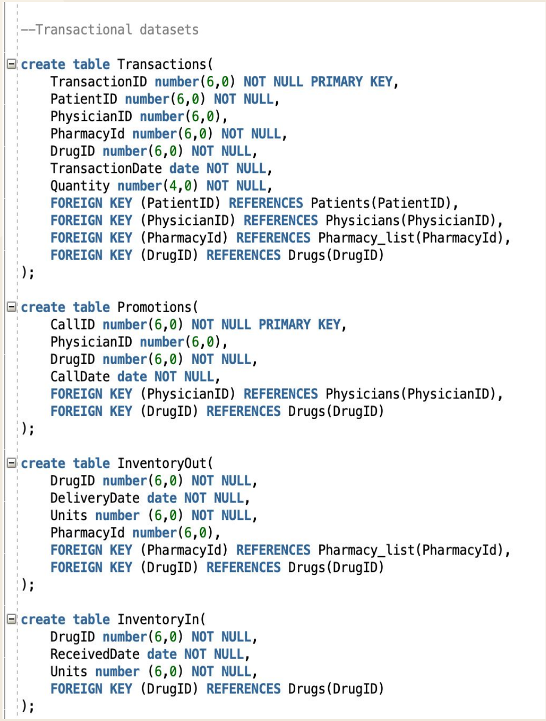
DDL Code

Core Tables



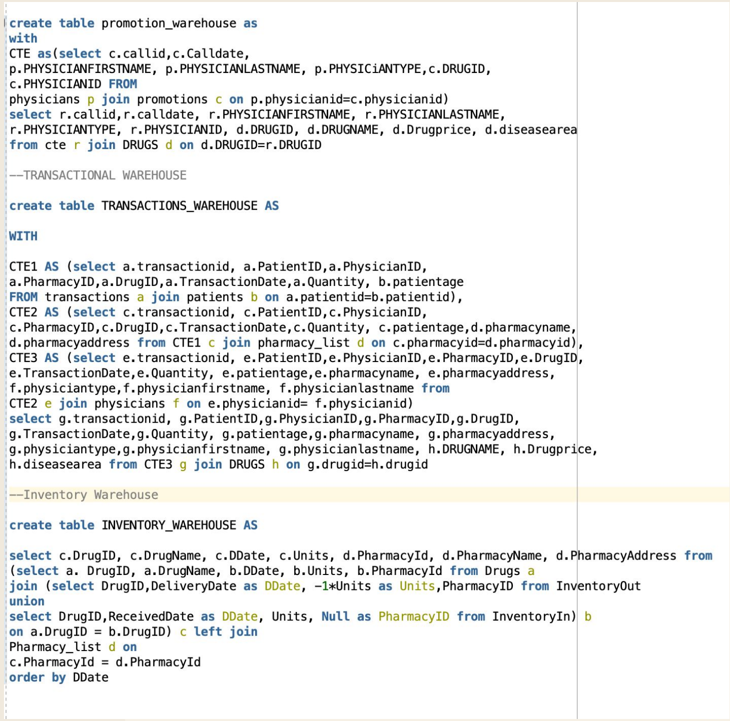
```sql
--Creating base tables with relational integrity

create table Physicians(
    PhysicianID number(6,0) NOT NULL PRIMARY KEY,
    PhysicianType varchar2(255),
    PhysicianFirstName varchar2(255),
    PhysicianLastName varchar2(255)
);

create table Patients(
    PatientID number(6,0) NOT NULL PRIMARY KEY,
    PatientAge varchar2(255)
);

create table Drugs(
    DrugID number(6,0) NOT NULL PRIMARY KEY,
    DrugName varchar2(255),
    DrugPrice number(5,0),
    DiseaseArea varchar2(255)
);

create table Pharmacy_list(
    PharmacyId number(6,0) not null Primary Key,
    PharmacyName Varchar2(255),
    PharmacyAddress Varchar2(255)
);
```

Transactional Tables



```sql
--Transactional datasets

create table Transactions(
    TransactionID number(6,0) NOT NULL PRIMARY KEY,
    PatientID number(6,0) NOT NULL,
    PhysicianID number(6,0),
    PharmacyId number(6,0) NOT NULL,
    DrugID number(6,0) NOT NULL,
    TransactionDate date NOT NULL,
    Quantity number(4,0) NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
    FOREIGN KEY (PharmacyId) REFERENCES Pharmacy_list(PharmacyId),
    FOREIGN KEY (DrugID) REFERENCES Drugs(DrugID)
);

create table Promotions(
    CallID number(6,0) NOT NULL PRIMARY KEY,
    PhysicianID number(6,0),
    DrugID number(6,0) NOT NULL,
    CallDate date NOT NULL,
    FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
    FOREIGN KEY (DrugID) REFERENCES Drugs(DrugID)
);

create table InventoryOut(
    DrugID number(6,0) NOT NULL,
    DeliveryDate date NOT NULL,
    Units number (6,0) NOT NULL,
    PharmacyId number(6,0),
    FOREIGN KEY (PharmacyId) REFERENCES Pharmacy_list(PharmacyId),
    FOREIGN KEY (DrugID) REFERENCES Drugs(DrugID)
);

create table InventoryIn(
    DrugID number(6,0) NOT NULL,
    ReceivedDate date NOT NULL,
    Units number (6,0) NOT NULL,
    FOREIGN KEY (DrugID) REFERENCES Drugs(DrugID)
);
```

Warehouse Tables



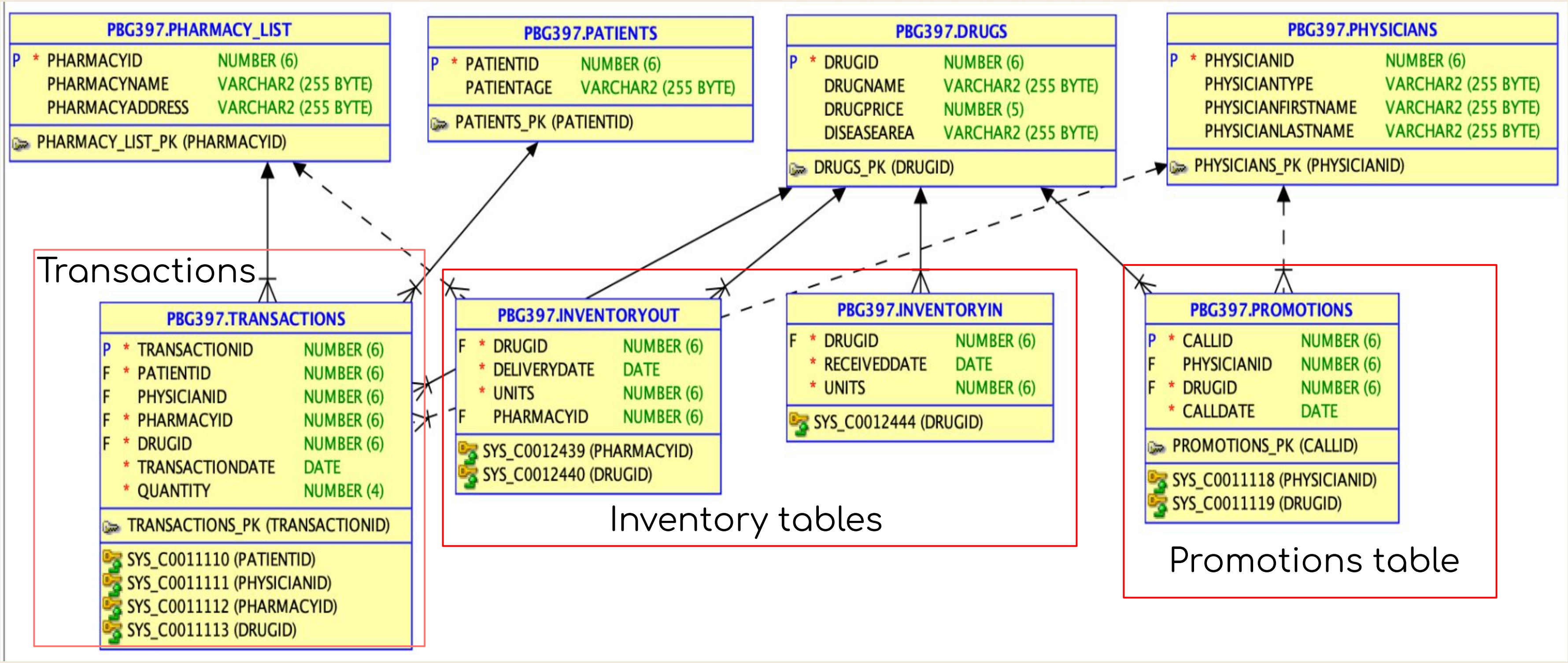
```sql
create table promotion_warehouse as
with
CTE as(select c.callid,c.Calldate,
p.PHYSICIANFIRSTNAME, p.PHYSICIANLASTNAME, p.PHYSICiANTYPE,c.DRUGID,
c.PHYSICIANID FROM
physicians p join promotions c on p.physicianid=c.physicianid)
select r.callid,r.calldate, r.PHYSICIANFIRSTNAME, r.PHYSICIANLASTNAME,
r.PHYSICIANTYPE, r.PHYSICIANID, d.DRUGID, d.DRUGNAME, d.Drugprice, d.diseasearea
from cte r join DRUGS d on d.DRUGID=r.DRUGID

--TRANSACTIONAL WAREHOUSE

create table TRANSACTIONS_WAREHOUSE AS

WITH

CTE1 AS (select a.transactionid, a.PatientID,a.PhysicianID,
a.PharmacyID,a.DrugID,a.TransactionDate,a.Quantity, b.patientage
FROM transactions a join patients b on a.patientid=b.patientid),
CTE2 AS (select c.transactionid, c.PatientID,c.PhysicianID,
c.PharmacyID,c.DrugID,c.TransactionDate,c.Quantity, c.patientage,d.pharmacyname,
d.pharmacyaddress from CTE1 c join pharmacy_list d on c.pharmacyid=d.pharmacyid),
CTE3 AS (select e.transactionid, e.PatientID,e.PhysicianID,e.PharmacyID,e.DrugID,
e.TransactionDate,e.Quantity, e.patientage,e.pharmacyname, e.pharmacyaddress,
f.physiciantype,f.physicianfirstname, f.physicianlastname from
CTE2 e join physicians f on e.physicianid= f.physicianid)
select g.transactionid, g.PatientID,g.PhysicianID,g.PharmacyID,g.DrugID,
g.TransactionDate,g.Quantity, g.patientage,g.pharmacyname, g.pharmacyaddress,
g.physiciantype,g.physicianfirstname, g.physicianlastname, h.DRUGNAME, h.Drugprice,
h.diseasearea from CTE3 g join DRUGS h on g.drugid=h.drugid

--Inventory Warehouse

create table INVENTORY_WAREHOUSE AS

select c.DrugID, c.DrugName, c.DDate, c.Units, d.PharmacyId, d.PharmacyName, d.PharmacyAddress from
(select a. DrugID, a.DrugName, b.DDate, b.Units, b.PharmacyId from Drugs a
join (select DrugID,DeliveryDate as DDate, -1*Units as Units,PharmacyID from InventoryOut
union
select DrugID,ReceivedDate as DDate, Units, Null as PharmacyID from InventoryIn) b
on a.DrugID = b.DrugID) c left join
Pharmacy_list d on
c.PharmacyId = d.PharmacyId
order by DDate
```

03

Data Models

Transactional Tables

Transactional Tables



PBG397.PHARMACY_LIST

P	*	PHARMACYID	NUMBER (6)
		PHARMACYNAME	VARCHAR2 (255 BYTE)
		PHARMACYADDRESS	VARCHAR2 (255 BYTE)

PHARMACY_LIST_PK (PHARMACYID)

PBG397.PATIENTS

P	*	PATIENTID	NUMBER (6)
		PATIENTAGE	VARCHAR2 (255 BYTE)

PATIENTS_PK (PATIENTID)

PBG397.DRUGS

P	*	DRUGID	NUMBER (6)
		DRUGNAME	VARCHAR2 (255 BYTE)
		DRUGPRICE	NUMBER (5)
		DISEASEAREA	VARCHAR2 (255 BYTE)

DRUGS_PK (DRUGID)

PBG397.PHYSICIANS

P	*	PHYSICIANID	NUMBER (6)
		PHYSICIANTYPE	VARCHAR2 (255 BYTE)
		PHYSICIANFIRSTNAME	VARCHAR2 (255 BYTE)
		PHYSICIANLASTNAME	VARCHAR2 (255 BYTE)

PHYSICIANS_PK (PHYSICIANID)

PBG397.TRANSACTIONS

P	*	TRANSACTIONID	NUMBER (6)
F	*	PATIENTID	NUMBER (6)
F		PHYSICIANID	NUMBER (6)
F	*	PHARMACYID	NUMBER (6)
F	*	DRUGID	NUMBER (6)
	*	TRANSACTIONDATE	DATE
	*	QUANTITY	NUMBER (4)

TRANSACTIONS_PK (TRANSACTIONID)

SYS_C0011110 (PATIENTID)
SYS_C0011111 (PHYSICIANID)
SYS_C0011112 (PHARMACYID)
SYS_C0011113 (DRUGID)

PBG397.INVENTORYOUT

F	*	DRUGID	NUMBER (6)
	*	DELIVERYDATE	DATE
	*	UNITS	NUMBER (6)
F		PHARMACYID	NUMBER (6)

SYS_C0012439 (PHARMACYID)
SYS_C0012440 (DRUGID)

PBG397.INVENTORYIN

F	*	DRUGID	NUMBER (6)
	*	RECEIVEDDATE	DATE
	*	UNITS	NUMBER (6)

SYS_C0012444 (DRUGID)

PBG397.PROMOTIONS

P	*	CALLID	NUMBER (6)
F		PHYSICIANID	NUMBER (6)
F	*	DRUGID	NUMBER (6)
	*	CALLDATE	DATE

PROMOTIONS_PK (CALLID)

SYS_C0011118 (PHYSICIANID)
SYS_C0011119 (DRUGID)

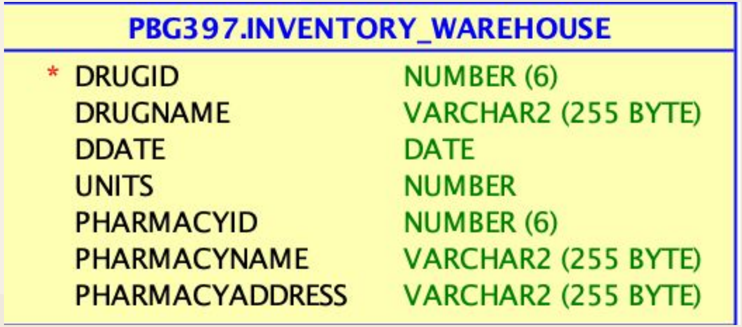
- Consists of the core tables at the top - Pharmacy_list, Patients, Drugs, Physicians
- Verifies referential integrity constraints while inserting data

04

Data Warehouse

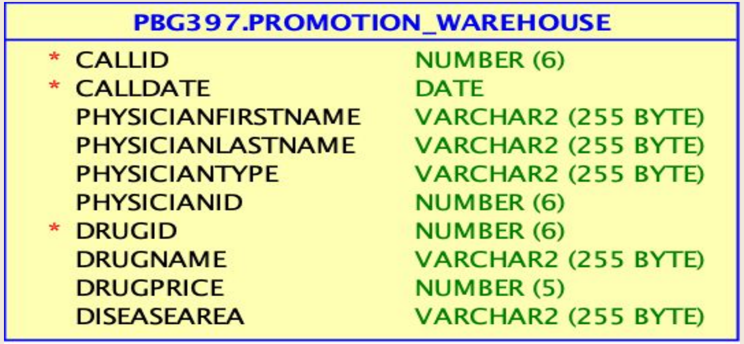
SSOT

SSOT



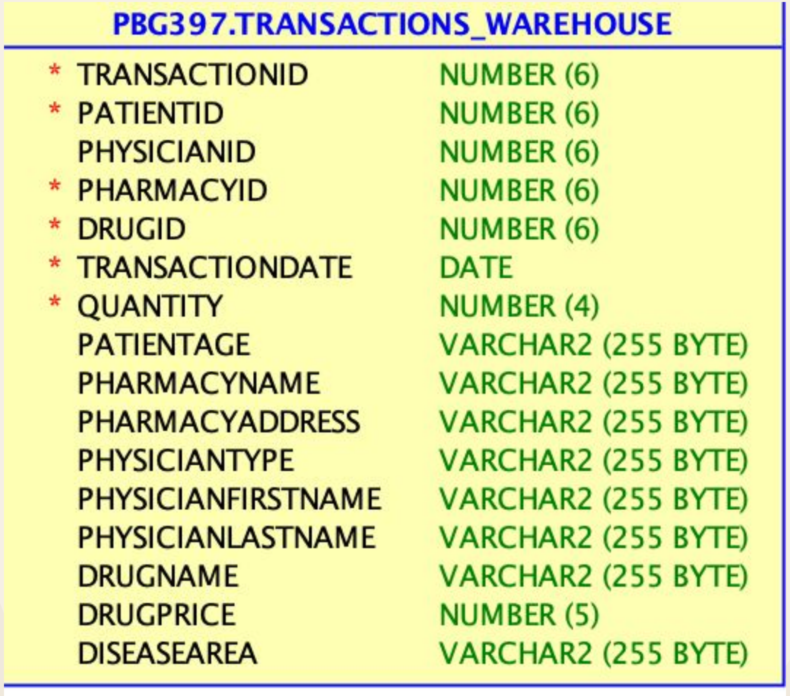
PBG397.INVENTORY_WAREHOUSE

*	DRUGID	NUMBER (6)
	DRUGNAME	VARCHAR2 (255 BYTE)
	DDATE	DATE
	UNITS	NUMBER
	PHARMACYID	NUMBER (6)
	PHARMACYNAME	VARCHAR2 (255 BYTE)
	PHARMACYADDRESS	VARCHAR2 (255 BYTE)



PBG397.PROMOTION_WAREHOUSE

*	CALLID	NUMBER (6)
*	CALLDATE	DATE
	PHYSICIANFIRSTNAME	VARCHAR2 (255 BYTE)
	PHYSICIANLASTNAME	VARCHAR2 (255 BYTE)
	PHYSICIANTYPE	VARCHAR2 (255 BYTE)
	PHYSICIANID	NUMBER (6)
*	DRUGID	NUMBER (6)
	DRUGNAME	VARCHAR2 (255 BYTE)
	DRUGPRICE	NUMBER (5)
	DISEASEAREA	VARCHAR2 (255 BYTE)



PBG397.TRANSACTIONS_WAREHOUSE

*	TRANSACTIONID	NUMBER (6)
*	PATIENTID	NUMBER (6)
	PHYSICIANID	NUMBER (6)
*	PHARMACYID	NUMBER (6)
*	DRUGID	NUMBER (6)
*	TRANSACTIONDATE	DATE
*	QUANTITY	NUMBER (4)
	PATIENTAGE	VARCHAR2 (255 BYTE)
	PHARMACYNAME	VARCHAR2 (255 BYTE)
	PHARMACYADDRESS	VARCHAR2 (255 BYTE)
	PHYSICIANTYPE	VARCHAR2 (255 BYTE)
	PHYSICIANFIRSTNAME	VARCHAR2 (255 BYTE)
	PHYSICIANLASTNAME	VARCHAR2 (255 BYTE)
	DRUGNAME	VARCHAR2 (255 BYTE)
	DRUGPRICE	NUMBER (5)
	DISEASEAREA	VARCHAR2 (255 BYTE)

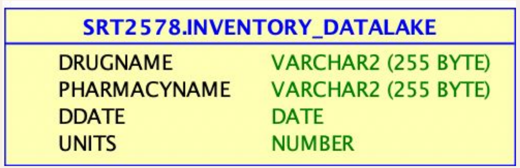
- SSOT for Novartis Healthcare
- Constructed by using data present in the transactional tables
- Data required for any analytical operations can be extracted using these tables

05

DataMarts

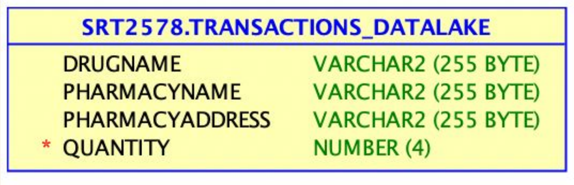
MVOT

MVOT



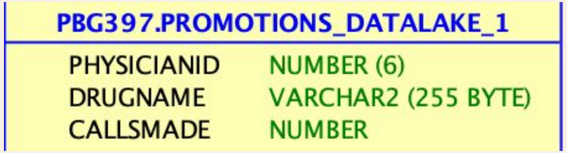
SRT2578.INVENTORY_DATALAKE

	DRUGNAME	VARCHAR2 (255 BYTE)
	PHARMACYNAME	VARCHAR2 (255 BYTE)
	DDATE	DATE
	UNITS	NUMBER



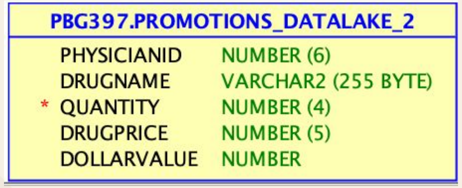
SRT2578.TRANSACTIONS_DATALAKE

	DRUGNAME	VARCHAR2 (255 BYTE)
	PHARMACYNAME	VARCHAR2 (255 BYTE)
	PHARMACYADDRESS	VARCHAR2 (255 BYTE)
*	QUANTITY	NUMBER (4)



PBG397.PROMOTIONS_DATALAKE_1

	PHYSICIANID	NUMBER (6)
	DRUGNAME	VARCHAR2 (255 BYTE)
	CALLSMADE	NUMBER



PBG397.PROMOTIONS_DATALAKE_2

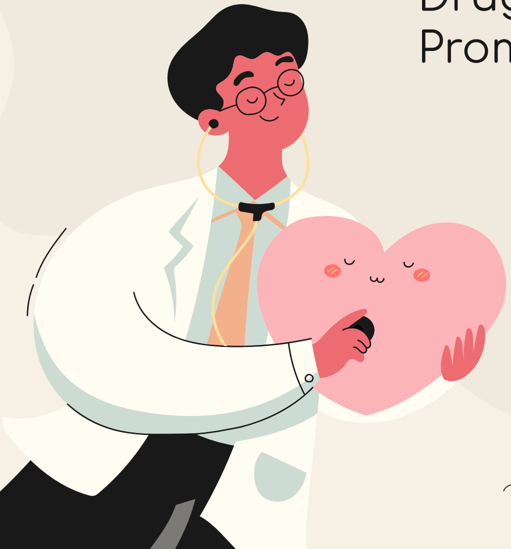
	PHYSICIANID	NUMBER (6)
	DRUGNAME	VARCHAR2 (255 BYTE)
*	QUANTITY	NUMBER (4)
	DRUGPRICE	NUMBER (5)
	DOLLARVALUE	NUMBER

- Datamarts are MVOTs derived from the SSOT - Warehouse tables
- These are relevant columns required to perform analytics for specific applications and support decision making
- Smaller number of columns and specific permissions in AWS S3 support fast access and reduces redundancy by removing irrelevant columns for the analysis
- Suffixed with ‘datalake’ as these form the csv’s stored in the datalake for Novartis

06

Analytics

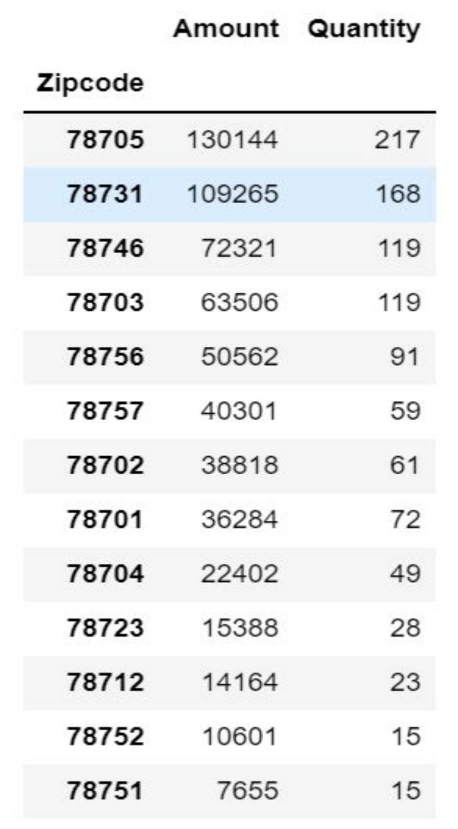
Region wise Analysis
Drug sales Analysis
Inventory Management
Drug return Analysis
Promotion Analysis



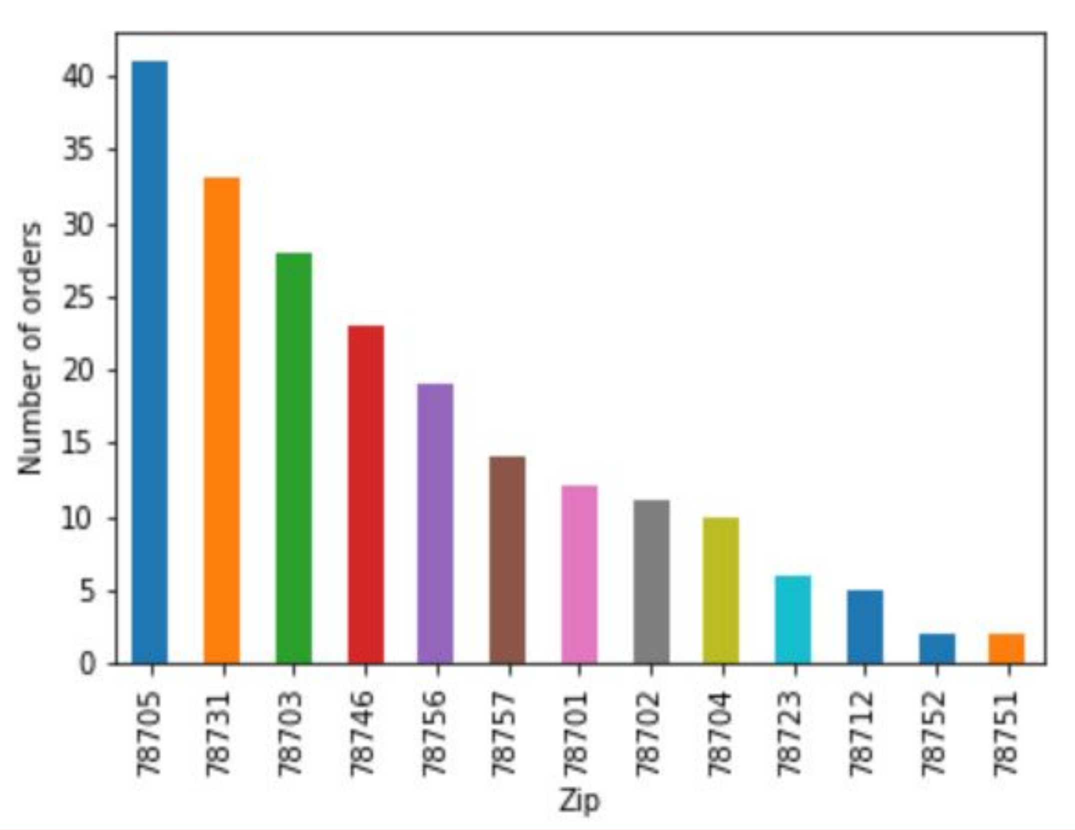
What are we looking for?

- How are the sales looking for different drugs and different regions?
- Are specific drugs more frequently prescribed in certain areas? This would allow us to understand if certain diseases are more common in an area
- Do we have an efficient inventory system? Are we meeting the needs of the customers?
- Are our promotional campaigns working?

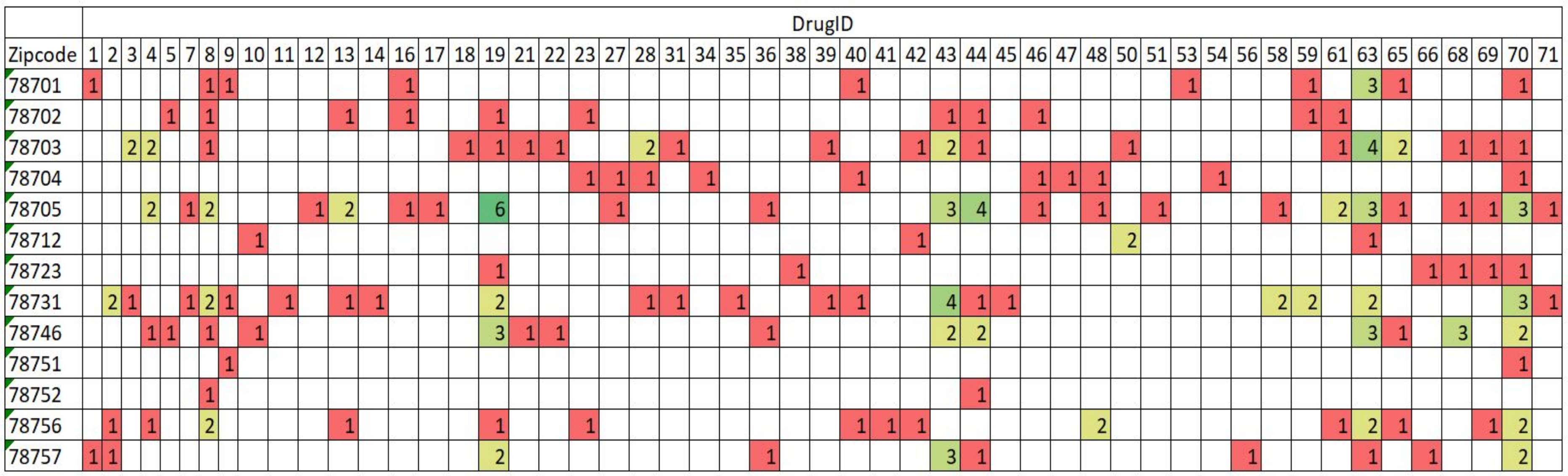
Region Wise Analysis



	Amount	Quantity
Zipcode		
78705	130144	217
78731	109265	168
78746	72321	119
78703	63506	119
78756	50562	91
78757	40301	59
78702	38818	61
78701	36284	72
78704	22402	49
78723	15388	28
78712	14164	23
78752	10601	15
78751	7655	15

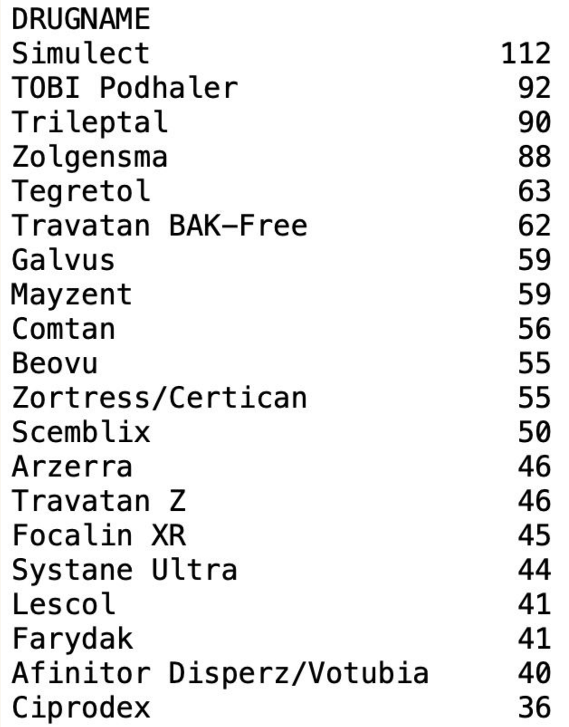


Drug Sales



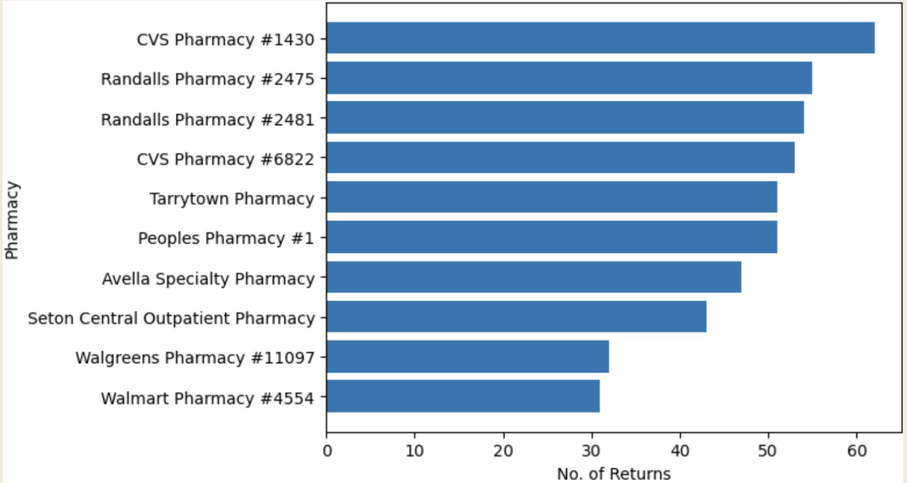
	DrugID																																																		
Zipcode	1	2	3	4	5	7	8	9	10	11	12	13	14	16	17	18	19	21	22	23	27	28	31	34	35	36	38	39	40	41	42	43	44	45	46	47	48	50	51	53	54	56	58	59	61	63	65	66	68	69	70	71
78701	1						1	1						1															1											1				1		3	1				1	
78702					1		1					1		1			1			1												1	1		1									1	1							
78703			2	2			1									1	1	1	1			2	1					1			1	2	1					1							1	4	2		1	1	1	
78704																				1	1	1		1					1						1	1	1				1										1	
78705				2		1	2				1	2		1	1		6				1					1						3	4		1		1		1				1		2	3	1		1	1	3	1
78712									1																						1							2								1						
78723																	1										1																					1	1	1	1	
78731		2	1			1	2	1		1		1	1				2					1	1		1			1	1			4	1	1									2	2		2					3	1
78746				1	1		1		1								3	1	1							1						2	2													3	1		3		2	
78751								1																																											1	
78752							1																										1																			
78756		1		1			2					1					1			1									1	1	1						2								1	2	1			1	2	
78757	1	1															2									1						3	1									1				1		1			2	

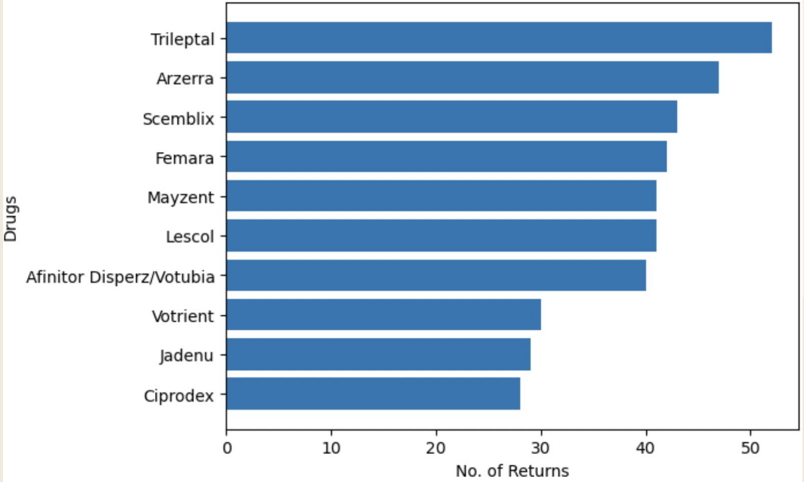
Inventory Management



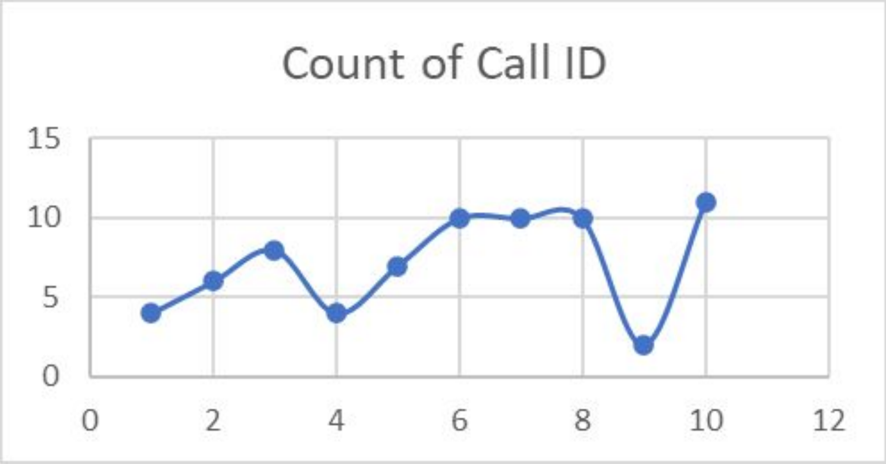
```
DRUGNAME
Simulect                    112
TOBI Podhaler                92
Trileptal                    90
Zolgensma                    88
Tegretol                     63
Travatan BAK-Free            62
Galvus                       59
Mayzent                      59
Comtan                       56
Beovu                        55
Zortress/Certican            55
Scemblix                     50
Arzerra                      46
Travatan Z                   46
Focalin XR                   45
Systane Ultra                44
Lescol                       41
Farydak                      41
Afinitor Disperz/Votubia     40
Ciprodex                     36
```

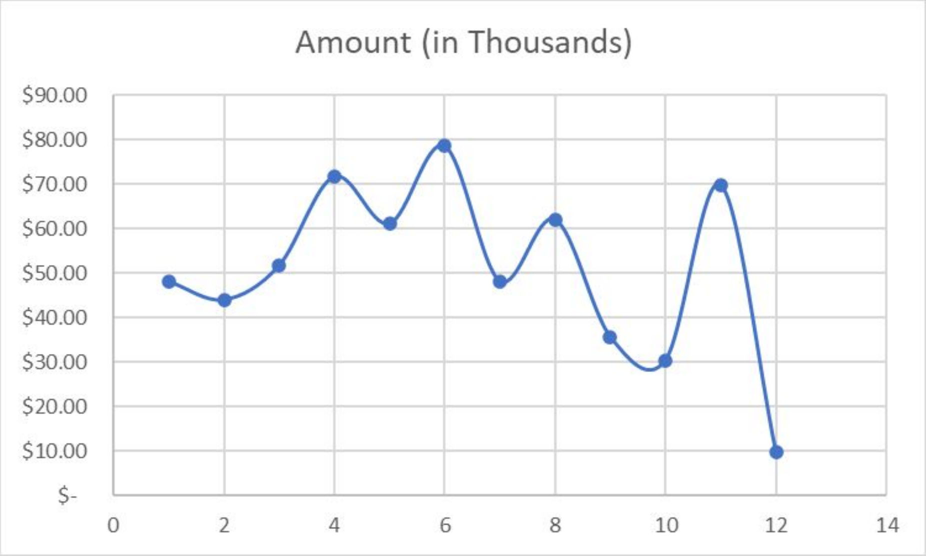
Drug Return Analysis



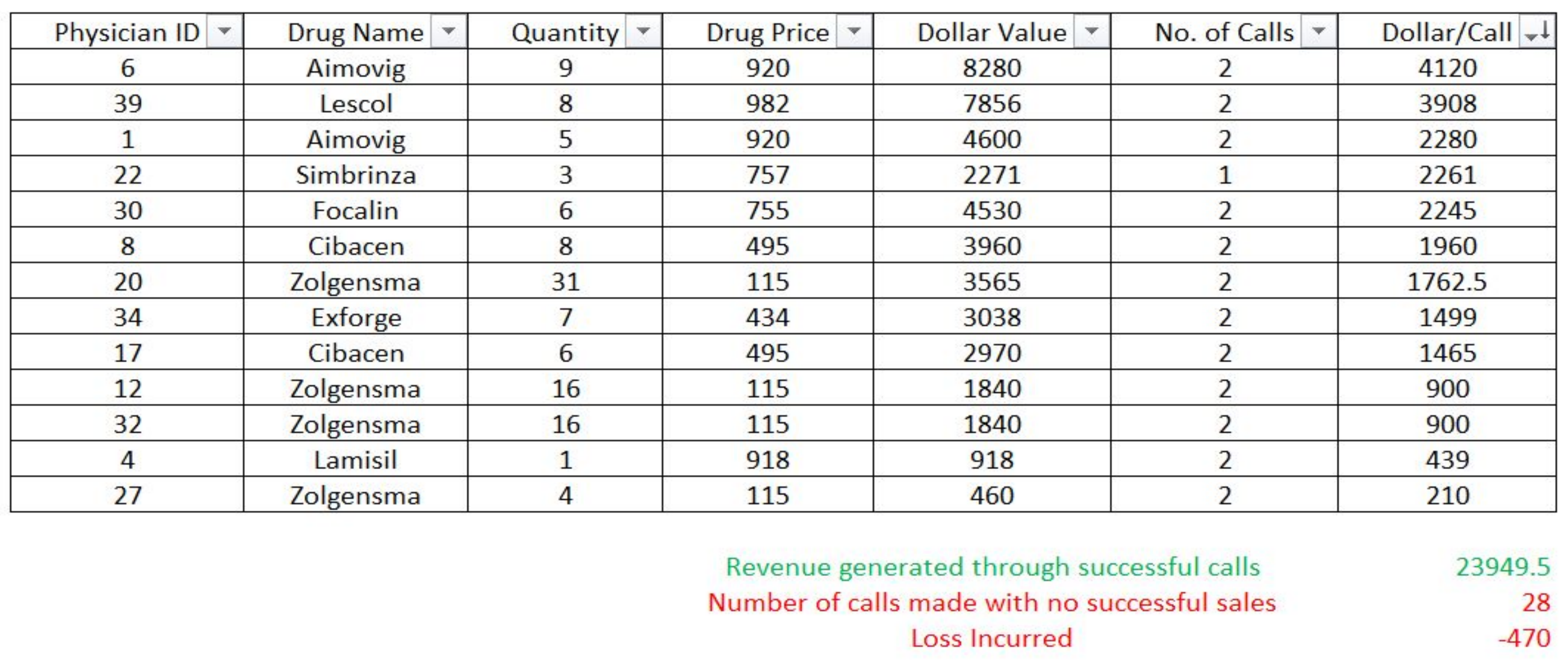


Promotion Analysis - Trends





Promotion Analysis - Revenues and Losses



Physician ID	Drug Name	Quantity	Drug Price	Dollar Value	No. of Calls	Dollar/Call
6	Aimovig	9	920	8280	2	4120
39	Lescol	8	982	7856	2	3908
1	Aimovig	5	920	4600	2	2280
22	Simbrinza	3	757	2271	1	2261
30	Focalin	6	755	4530	2	2245
8	Cibacen	8	495	3960	2	1960
20	Zolgensma	31	115	3565	2	1762.5
34	Exforge	7	434	3038	2	1499
17	Cibacen	6	495	2970	2	1465
12	Zolgensma	16	115	1840	2	900
32	Zolgensma	16	115	1840	2	900
4	Lamisil	1	918	918	2	439
27	Zolgensma	4	115	460	2	210

Revenue generated through successful calls	23949.5
Number of calls made with no successful sales	28
Loss Incurred	-470

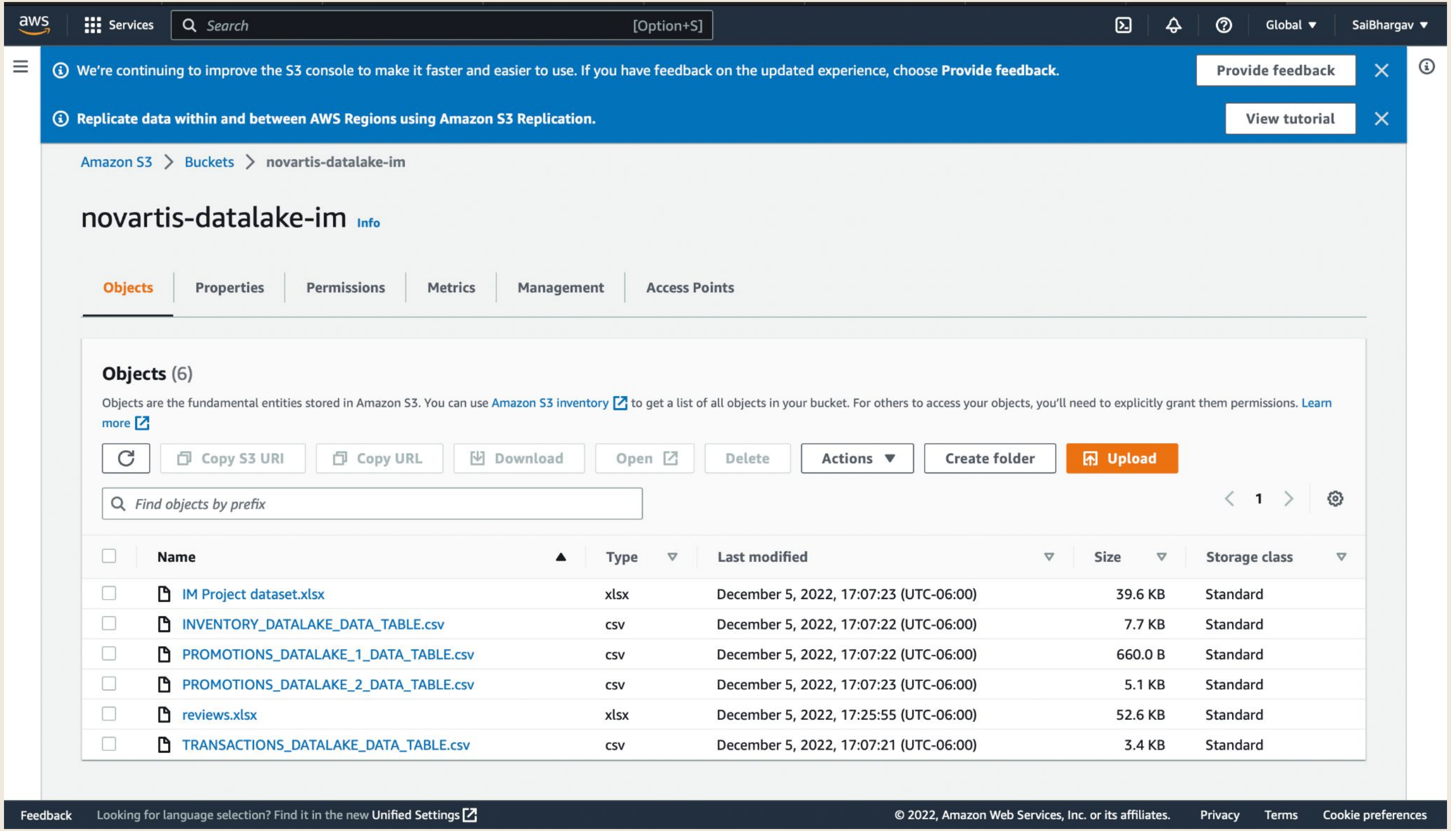
What did we learn?

- Oracle SQL developer - Setting up the database
- How basic healthcare pharma company information systems work
- Importance of warehouse (SSOT) in healthcare
- Performing analysis while maintaining data privacy

Thank you!

Any questions?

Appendix



We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Replicate data within and between AWS Regions using Amazon S3 Replication.

novartis-datalake-im Info

Objects (6)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	IM Project dataset.xlsx	xlsx	December 5, 2022, 17:07:23 (UTC-06:00)	39.6 KB	Standard
□	INVENTORY_DATALAKE_DATA_TABLE.csv	csv	December 5, 2022, 17:07:22 (UTC-06:00)	7.7 KB	Standard
□	PROMOTIONS_DATALAKE_1_DATA_TABLE.csv	csv	December 5, 2022, 17:07:22 (UTC-06:00)	660.0 B	Standard
□	PROMOTIONS_DATALAKE_2_DATA_TABLE.csv	csv	December 5, 2022, 17:07:23 (UTC-06:00)	5.1 KB	Standard
□	reviews.xlsx	xlsx	December 5, 2022, 17:25:55 (UTC-06:00)	52.6 KB	Standard
□	TRANSACTIONS_DATALAKE_DATA_TABLE.csv	csv	December 5, 2022, 17:07:21 (UTC-06:00)	3.4 KB	Standard

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.