Purdue University

Pfizer Analytics Group

Isabella Alvarado
Michael Gospodarek
Utkarsh Singh
Jim Lin

Database Management System MGMT 544
Dr. Jerry DeHondt

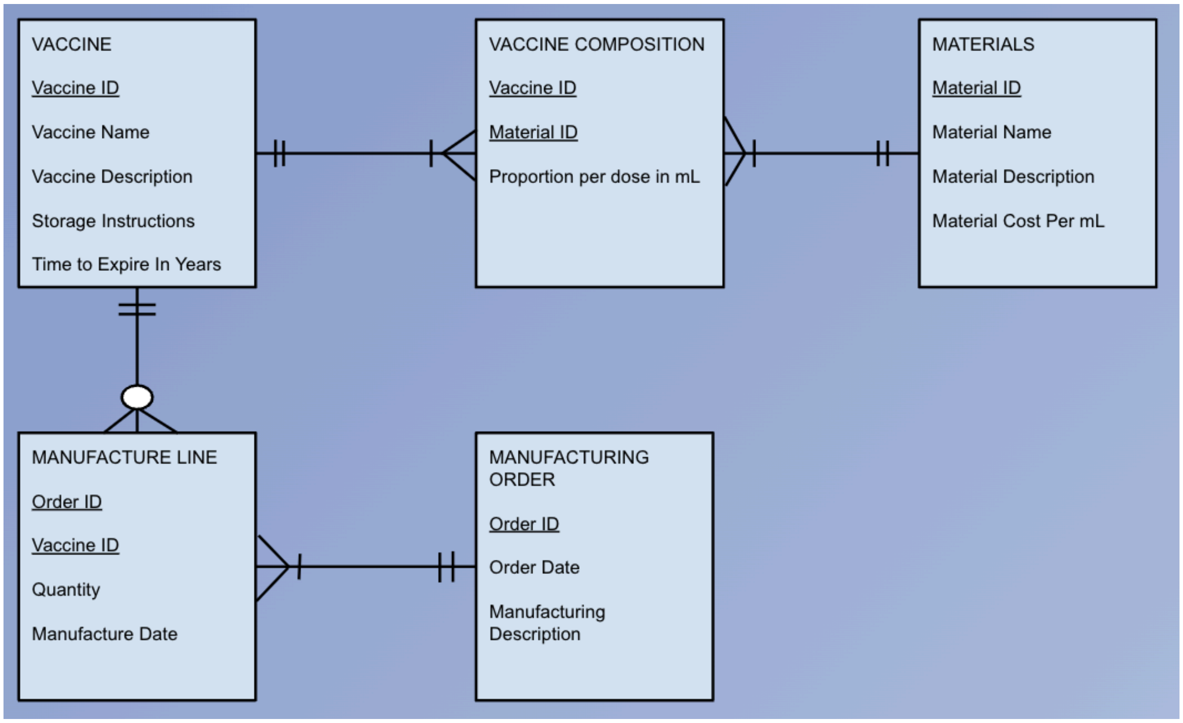
Background of Pfizer Analytics Group:

We are a team formed by Pfizer Inc. to better help them gain a better understanding of the company through data analytics. Pfizer is a multinational pharmaceutical and biotechnology corporation. Pfizer focuses on different products that can potentially save millions of lives. For Pfizer to have a breakthrough in the medical industry, they need data to help them achieve their goals. Currently, we are focused on the manufacturing part of the company. The manufacturing issues that arose during COVID has left many, throughout the world, without the chance of getting a vaccine. As a team we are trying to find information on how we can prevent this from happening again. How do we lower costs and keep up with the demand at the same time? What is the demand for a specific vaccine? Which vaccines produce the most profit? We want to help Pfizer with figuring out which parts of the manufacturing line bring the most value to the company.

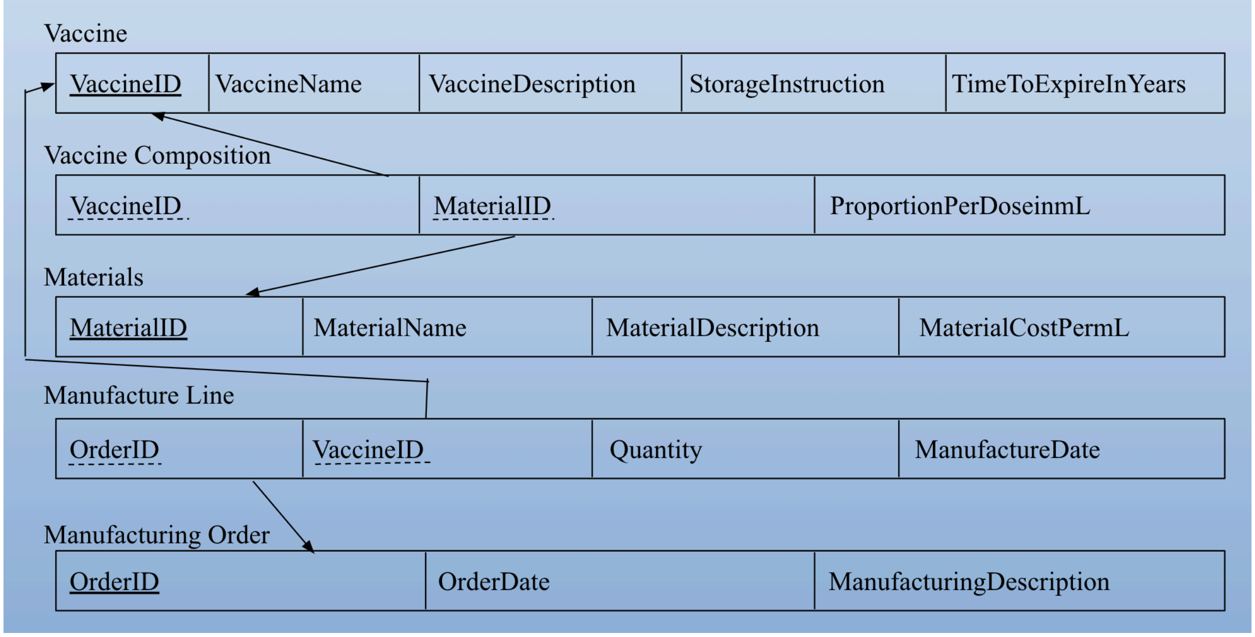
Project Objectives:

We are focused on using data to help Pfizer plan for production and gain a better understanding of their cost. Pfizer produces many different vaccines and we are tasked to determine which one will bring us the most profit, how many orders we have, etc. We useSQL to find out questions such as “What are the top 3 most ordered vaccines?” to predict and plan for future demands.

ERD Diagram



Relational Schema



Vaccine

VaccineID	VaccineName	VaccineDescription	StorageInstruction	TimeToExpireInYears

Vaccine Composition

VaccineID	MaterialID	ProportionPerDoseinmL

Materials

MaterialID	MaterialName	MaterialDescription	MaterialCostPermL

Manufacture Line

OrderID	VaccineID	Quantity	ManufactureDate

Manufacturing Order

OrderID	OrderDate	ManufacturingDescription

Data Model and Design Choices

This model is intended to minimize redundancy for storage purposes. Storing information about order quantities and material costs allows us to understand demand patterns and associated cost spikes.

Important information is stored in the Vaccine table alongside the VaccineID primary key. Such information is not necessary to report or store with every required material for each vaccine. Splitting Vaccine from its composition this way mitigates data redundancy.

Vaccine composition is the combination of vaccine and material IDs interacting to understand the makeup of each vaccine. It would have been cumbersome to store many instances of a vaccine's information to understand its makeup - this merits Vaccine Composition to exist as its own table. It has a one (mandatory) to many (optional) relationship, as a given vaccine is not included in every order.

Materials exist as its own table for similar reasons to vaccines, that is to reduce redundancy and exist in 3rd normal form from the Vaccine Composition table. Materials have a one to many relationship into vaccine composition since one material could be a component of many different vaccines.

Manufacture Line is another key component of our model that stores important information of the orders requested from this Vaccine company, but also does so in an efficient way. It is unnecessary to store every piece of vaccine or material information alongside each manufactured order, thus inspiring why it's its own table.

Finally, manufacturing order exists as its own table sensibly as a means to keep a record of order dates with minimal unnecessary information attached, but accessible if needed.

The design choices were intended to achieve third normal form in reducing redundancy.

Question/ Description of Queries:

1. What are the Storage Instructions for the vaccine called “Prevenar13”?
   The following query will show the storage instruction for the vaccine called “Prevenar13”

   ```
   SELECT StorageInstructions
   FROM VACCINES
   WHERE  VaccineName = ‘Prevenar13’;
   ```

   This query will show employees’ of Pfizer how to store the vaccine “Prevenar13”.

2. What is the vaccine with the highest ordered quantity?
   The following query will show the vaccine with the highest ordered quantity

   ```
   SELECT VaccineName, MAX(Quantity) as Quantity
   FROM VACCINES V, MANUFACTURELINE ML
   WHERE V.VaccineID = ML.VaccineID;
   ```

   Production planner can use this query to delegate extra space for the vaccine.

3. Find the proportion per Dose for “JE” Material ID.
   The following query will show the proportion per dose for material “JE”

   ```
   SELECT VaccineID, PorportionPerDoseinmL
   FROM VACCINECOMPOSITION
   WHERE VaccineID = 'JE';
   ```

   The query will be used by manufacturers to find out the proper dosage for “JE”.

4. As a comparison of expiration date after vaccine production. How many years would the oldest vaccine expire?
   The following query will show the expiration time for the vaccine with the shortest amount of shelf life

   ```
   SELECT VaccineName, MIN(TimeToExpireInYears) as ShortestTimetoExpire
   FROM VACCINES;
   ```

   The query will help the pharmaceutical be aware of how much to order with a shorter shelf life.

5. The Product Portfolio Manager wants a list of all vaccines with an order quantity higher than 500,000.
   The following query will show all vaccine with order quantity greater than 500,000

   Method 1: inner join

   ```
   SELECT VaccineName, Quantity
   FROM VACCINES AS A
   INNER JOIN MANUFACTURELINE AS B ON A.VaccineID = B.VaccineID
   ```

```
WHERE Quantity > 500000
GROUP BY VaccineName
ORDER BY Quantity DESC;
```

Method 2: subquery

```
SELECT VaccineName, A.Quantity
FROM VACCINES, (
        SELECT VaccineID, Quantity
   FROM MANUFACTURELINE
   WHERE Quantity > 500000
   GROUP BY VaccineID) AS A
WHERE VACCINES.VaccineID = A.VaccineID
ORDER BY A.Quantity DESC;
```

The query will help production planner delegate production schedule based on quantity.

6. A Demand Planner wants a list of the top 3 most ordered vaccines in the Production Site

   The following query will show the top 3 most ordered vaccines

   ```
   SELECT VaccineName, SUM(Quantity)
   FROM VACCINES, MANUFACTURELINE
   WHERE VACCINES.VaccineID = MANUFACTURELINE.VaccineID
   GROUP BY VaccineName
   ORDER BY SUM(Quantity) DESC LIMIT 3;
   ```

   The query will help Pfizer understand which vaccines are in the highest demand and improve forecasts in inventory.

7. The Supply Chain Department is evaluating the costs of production. List the total material costs for each vaccine per dose..

   The following query will show the material cost for each vaccine

   ```
   SELECT VaccineName, SUM(MaterialCostPermL) AS 'Material Cost Per Dose'
   FROM MATERIALS, VACCINECOMPOSITION, VACCINES
   WHERE MATERIALS.MaterialID = VACCINECOMPOSITION.MaterialID
   AND VACCINECOMPOSITION.VaccineID = VACCINES.VaccineID
   GROUP BY VaccineName;
   ```

   The query will help Pfizer understand which vaccines cost the most or least to produce and how much they should produce to keep a positive profit.

8. The Supply Chain Manager wants to know the Manufacturing Description, Order Date and the average Quantity for each vaccine.

   The following query will show the manufacturing description, order date and average quantity for each vaccine

   ```
   SELECT ManufacturingDescription, OrderDate,
   ```

```
    (SELECT AVG(Quantity) FROM MANUFACTURELINE WHERE
VaccineID = B.VaccineID) AvgQuantity
FROM MANUFACTURINGORDER AS A
LEFT OUTER JOIN MANUFACTURELINE AS B
ON A.OrderID = B.OrderID
ORDER BY OrderDate;
```

The query will Pfizer in planning for storage room. The average quantity of each vaccine and when they are ordered can allow them to properly be prepared for the incoming vaccines.

9. Find the Vaccine Names with Proportion Dose of 0.1.

The following query will show all vaccine names with a proportion dose of 0.1

```
SELECT VaccineName, PorportionPerDoseinmL
FROM VACCINES, VACCINECOMPOSITION
WHERE VACCINES.VaccineID IN (SELECT VaccineID FROM
VACCINECOMPOSITION WHERE PorportionPerDoseinmL = 0.1)
AND VACCINES.VaccineID = VACCINECOMPOSITION.VaccineID
ORDER BY VaccineName;
```

The query will show Pfizer which vaccines needs to have a proportion dose of 0.1

10. The Product Portfolio Manager wants a list of all vaccines ordered in the second quarter of 2020.

The following query will show all vaccines ordered in the second quarter of 2020

```
SELECT VACCINES.VaccineName, A.OrderDate
FROM VACCINES, MANUFACTURELINE,
    (SELECT *
    FROM MANUFACTURINGORDER
    WHERE OrderDate >'2020-04-01'
    AND OrderDate < '2020-07-01') as A
WHERE VACCINES.VaccineID = MANUFACTURELINE.VaccineID
AND MANUFACTURELINE.OrderID = A.OrderID;
```

The query will help the manager understand trends in the market and prepare for the next quarter. It will also help to prepare for future demand and inventory control.

Executed to Text:

```
Execute:
> SELECT VaccineName, StorageInstructions
FROM VACCINES
WHERE  VaccineName = 'Prevenar13'

+ ---------------- + ------------------------ +
| VaccineName      | StorageInstructions      |
+ ---------------- + ------------------------ +
| Prevenar13       | Refrigerated between 2°C and 8°C (36°F and 46°F) in original packaging |
+ ---------------- + ------------------------ +
1 rows

Execute:
> SELECT VaccineName, MAX(Quantity) as Quantity
FROM VACCINES V, MANUFACTURELINE ML
WHERE V.VaccineID = ML.VaccineID

+ ---------------- + ------------- +
| VaccineName      | Quantity      |
+ ---------------- + ------------- +
| Human Papillomavirus | 22000000      |
+ ---------------- + ------------- +
1 rows

Execute:
> SELECT VaccineID, PorportionPerDoseinmL
FROM VACCINECOMPOSITION
WHERE VaccineID = 'JE'

+ -------------- + -------------------------- +
| VaccineID      | PorportionPerDoseinmL      |
+ -------------- + -------------------------- +
| JE             | 0.4                        |
+ -------------- + -------------------------- +
1 rows
```

```
Execute:
> SELECT VaccineName, MIN(TimeToExpireInYears) as ShortestTimetoExpire
FROM VACCINES

+ ---------------- + ------------------------- +
| VaccineName      | ShortestTimetoExpire      |
+ ---------------- + ------------------------- +
| Human Papillomavirus | 1                     |
+ ---------------- + ------------------------- +
1 rows

Execute:
> SELECT VaccineName, A.Quantity
FROM VACCINES, (
        SELECT VaccineID, Quantity
   FROM MANUFACTURELINE
   WHERE Quantity > 500000
   GROUP BY VaccineID) AS A
WHERE VACCINES.VaccineID = A.VaccineID
ORDER BY A.Quantity DESC

+ ---------------- + ------------- +
| VaccineName      | Quantity      |
+ ---------------- + ------------- +
| Hepatitis A      | 22000000      |
| M-M-R II         | 1000000       |
| Human Papillomavirus | 740000        |
| Polio            | 670000        |
| Hepatitis B      | 670000        |
| Hib              | 660000        |
| Influenza        | 590000        |
| Prevenar13       | 540000        |
| DTaP             | 540000        |
| Herpes Zoster    | 510000        |
+ ---------------- + ------------- +
10 rows
```

```
Execute:
> SELECT VaccineName, Quantity
FROM VACCINES AS A
INNER JOIN MANUFACTURELINE AS B ON A.VaccineID = B.VaccineID
WHERE Quantity > 500000
GROUP BY VaccineName
ORDER BY Quantity DESC

+ ---------------- + ------------- +
| VaccineName      | Quantity      |
+ ---------------- + ------------- +
| Hepatitis A      | 22000000      |
| M-M-R II         | 1000000       |
| Human Papillomavirus | 740000        |
| Hepatitis B      | 670000        |
| Polio            | 670000        |
| Hib              | 660000        |
| Influenza        | 590000        |
| DTaP             | 540000        |
| Prevenar13       | 540000        |
| Herpes Zoster    | 510000        |
+ ---------------- + ------------- +
10 rows
```

```
Execute:
> SELECT VaccineName, SUM(Quantity)
FROM VACCINES, MANUFACTURELINE
WHERE VACCINES.VaccineID = MANUFACTURELINE.VaccineID
GROUP BY VaccineName
ORDER BY SUM(Quantity) DESC LIMIT 3

+ ---------------- + ------------------ +
| VaccineName      | SUM(Quantity)      |
+ ---------------- + ------------------ +
| Hepatitis A      | 23650000           |
| M-M-R II         | 2330000            |
| DTaP             | 1670000            |
+ ---------------- + ------------------ +
3 rows
```

Execute:

```
> SELECT VaccineName, SUM(MaterialCostPermL) AS 'Material Cost Per Dose'
FROM MATERIALS, VACCINECOMPOSITION, VACCINES
WHERE MATERIALS.MaterialID = VACCINECOMPOSITION.MaterialID
AND VACCINECOMPOSITION.VaccineID = VACCINES.VaccineID
GROUP BY VaccineName

+ ---------------- + --------------------------- +
| VaccineName      | Material Cost Per Dose      |
+ ---------------- + --------------------------- +
| Antrax           | 0.98                        |
| DTaP             | 1.24                        |
| Hepatitis A      | 2.01                        |
| Hepatitis B      | 0.7                         |
| Herpes Zoster    | 0.82                        |
| Hib              | 0.82                        |
| Human Papillomavirus | 0.33                        |
| Influenza        | 0.98                        |
| Japanese Encephalitis | 0.54                        |
| M-M-R II         | 1.52                        |
| Meningococcal    | 0.82                        |
| Nimenrix         | 1.02                        |
| Pneumococcal     | 0.33                        |
| Polio            | 0.54                        |
| Prevenar13       | 3.36                        |
| YF-Vax           | 1.7400000000000002          |
+ ---------------- + --------------------------- +
16 rows
```

Execute:

```
> SELECT ManufacturingDescription, OrderDate,
        (SELECT AVG(Quantity) FROM MANUFACTURELINE WHERE VaccineID =
B.VaccineID) AvgQuantity
FROM MANUFACTURINGORDER AS A
LEFT OUTER JOIN MANUFACTURELINE AS B
ON A.OrderID = B.OrderID
ORDER BY OrderDate

+ ---------------------------- + ------------- + --------------- +
| ManufacturingDescription      | OrderDate      | AvgQuantity      |
+ ---------------------------- + ------------- + --------------- +
| Package in halves to ship separately | 2020-06-19 00:00:00 | 450000.0000      |
| Package in halves to ship separately | 2020-08-17 00:00:00 | 540000.0000      |
| Deliver to same address as last order | 2020-10-22 00:00:00 | 4730000.0000     |
| N/A                          | 2020-11-02 00:00:00 | 388333.3333      |
| N/A                          | 2020-12-09 00:00:00 | 388333.3333      |
| Package in halves to ship separately | 2021-02-28 00:00:00 | 345000.0000      |
| Deliver to same address as last order | 2021-06-21 00:00:00 | 450000.0000      |
| Early order                   | 2021-06-22 00:00:00 | 110000.0000      |
| N/A                          | 2021-08-14 00:00:00 | 670000.0000      |
| Deliver between 2 locations   | 2021-11-07 00:00:00 | 376666.6667      |
| For pickup                    | 2021-11-08 00:00:00 | 376666.6667      |
| For pickup                    | 2021-11-28 00:00:00 | 388333.3333      |
| N/A                          | 2022-01-14 00:00:00 | 515000.0000      |
| For pickup                    | 2022-01-15 00:00:00 | 415000.0000      |
| N/A                          | 2022-05-01 00:00:00 | 450000.0000      |
| For pickup                    | 2022-05-23 00:00:00 | 240000.0000      |
| Special delivery instructions: get 2 sign offs upon delivery | 2022-07-25 00:00:00 | 590000.0000
|
| Early order                   | 2022-07-30 00:00:00 | 490000.0000      |
| Requested delivery in 2 months | 2022-08-21 00:00:00 | 4730000.0000     |
| Requested delivery in 2 months | 2022-09-27 00:00:00 | 275000.0000      |
+ ---------------------------- + ------------- + --------------- +
20 rows
```

```
Execute:
> SELECT VaccineName, PorportionPerDoseinmL
FROM VACCINES, VACCINECOMPOSITION
WHERE VACCINES.VaccineID IN (SELECT VaccineID FROM VACCINECOMPOSITION
WHERE PorportionPerDoseinmL = 0.1)
AND VACCINES.VaccineID = VACCINECOMPOSITION.VaccineID
ORDER BY VaccineName

+ ---------------- + -------------------------- +
| VaccineName      | PorportionPerDoseinmL      |
+ ---------------- + -------------------------- +
| Herpes Zoster    | 0.1                        |
+ ---------------- + -------------------------- +
1 rows

Execute:
> SELECT VACCINES.VaccineName, A.OrderDate
FROM VACCINES, MANUFACTURELINE,
        (SELECT *
   FROM MANUFACTURINGORDER
   WHERE OrderDate > '2020-04-01'
        AND OrderDate < '2020-07-01') as A
WHERE VACCINES.VaccineID = MANUFACTURELINE.VaccineID
AND MANUFACTURELINE.OrderID = A.OrderID

+ ---------------- + -------------- +
| VaccineName      | OrderDate      |
+ ---------------- + -------------- +
| Hib              | 2020-06-19 00:00:00 |
+ ---------------- + -------------- +
1 rows
```