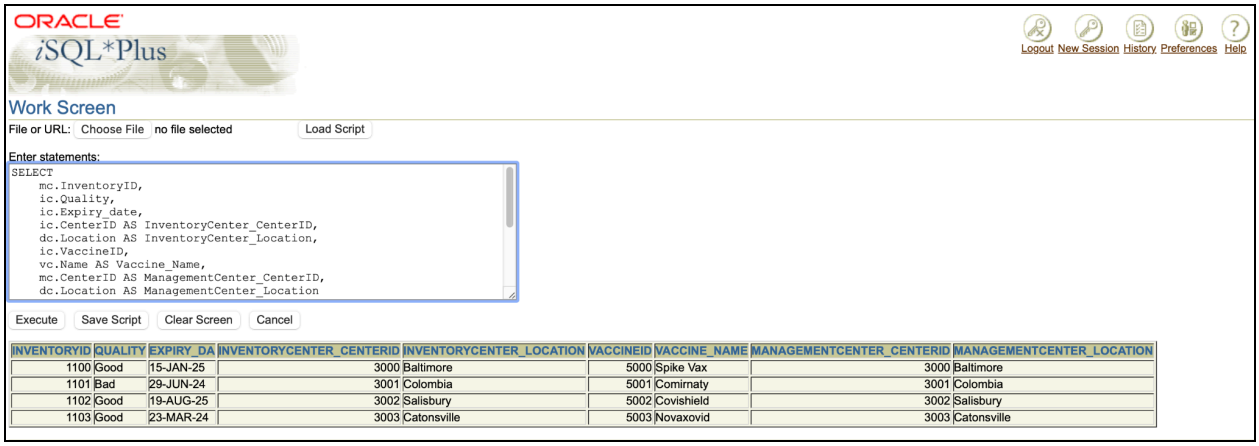
Project Deliverable - I

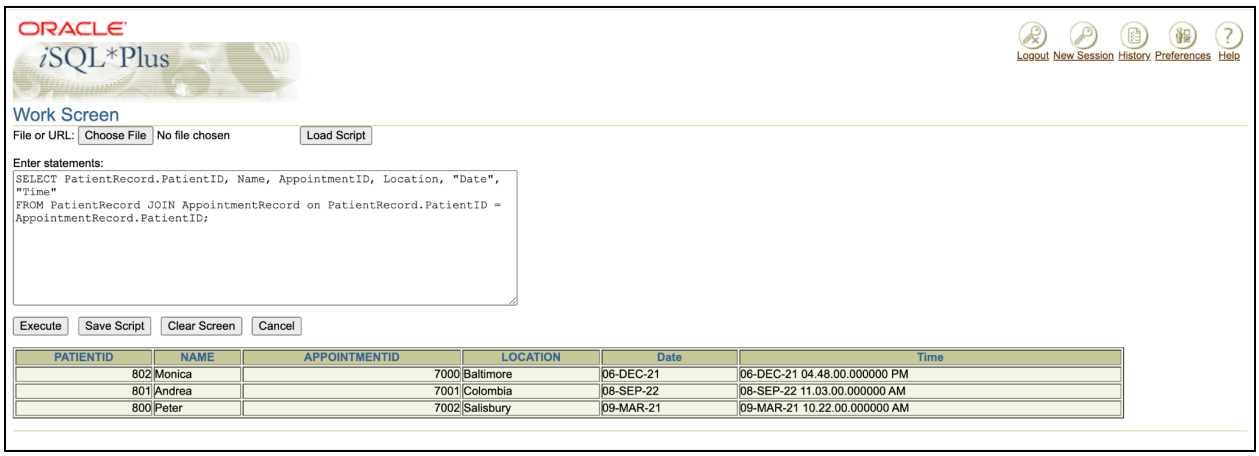
GROUP 6 Query Output:



INVENTORYID	QUALITY	EXPIRY_DA	INVENTORYCENTER_CENTERID	INVENTORYCENTER_LOCATION	VACCINEID	VACCINE_NAME	MANAGEMENTCENTER_CENTERID	MANAGEMENTCENTER_LOCATION
1100	Good	15-JAN-25	3000	Baltimore	5000	Spike Vax	3000	Baltimore
1101	Bad	29-JUN-24	3001	Colombia	5001	Comirnaty	3001	Colombia
1102	Good	19-AUG-25	3002	Salisbury	5002	Covishield	3002	Salisbury
1103	Good	23-MAR-24	3003	Catonsville	5003	Novaxovid	3003	Catonsville

Query: For each patient who places an appointment, what is the patient's name and ID as well as their Appointment ID, Location, Date and Time.

```
SELECT PatientRecord.PatientID, AppointmentID, Name, Location, "Date", "Time"
FROM PatientRecord JOIN AppointmentRecord on PatientRecord.PatientID =
AppointmentRecord.PatientID;
```



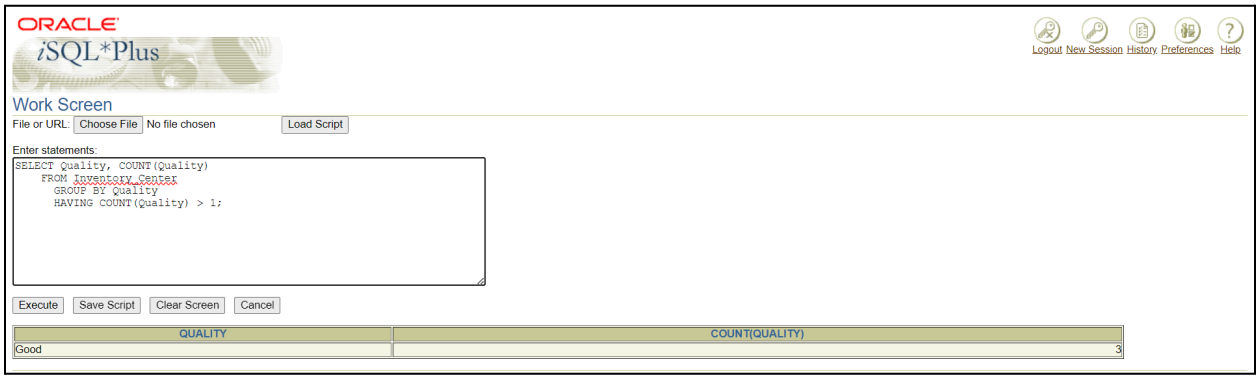
PATIENTID	NAME	APPOINTMENTID	LOCATION	Date	Time
802	Monica	7000	Baltimore	06-DEC-21	06-DEC-21 04.48.00.000000 PM
801	Andrea	7001	Colombia	08-SEP-22	08-SEP-22 11.03.00.000000 AM
800	Peter	7002	Salisbury	09-MAR-21	09-MAR-21 10.22.00.000000 AM

This query is used to to easily pull up the basic patient's appointment details. The query joins tables of the Patient Record and Appointment record. It shows every patient's ID along with their name. It then specifies their appointment ID, where it is located at, the specific date and time.

```
— - Group by total immunocompromised
SELECT Immunocompromised_status, COUNT(Immunocompromised_status)
    FROM PatientRecord
      GROUP BY Immunocompromised_status
        ORDER BY Immunocompromised_status;
```

This query will count the number of people who are immunocompromised. One column will display the immunocompromised status (yes or no) and the other column will show the number of people it applies to.

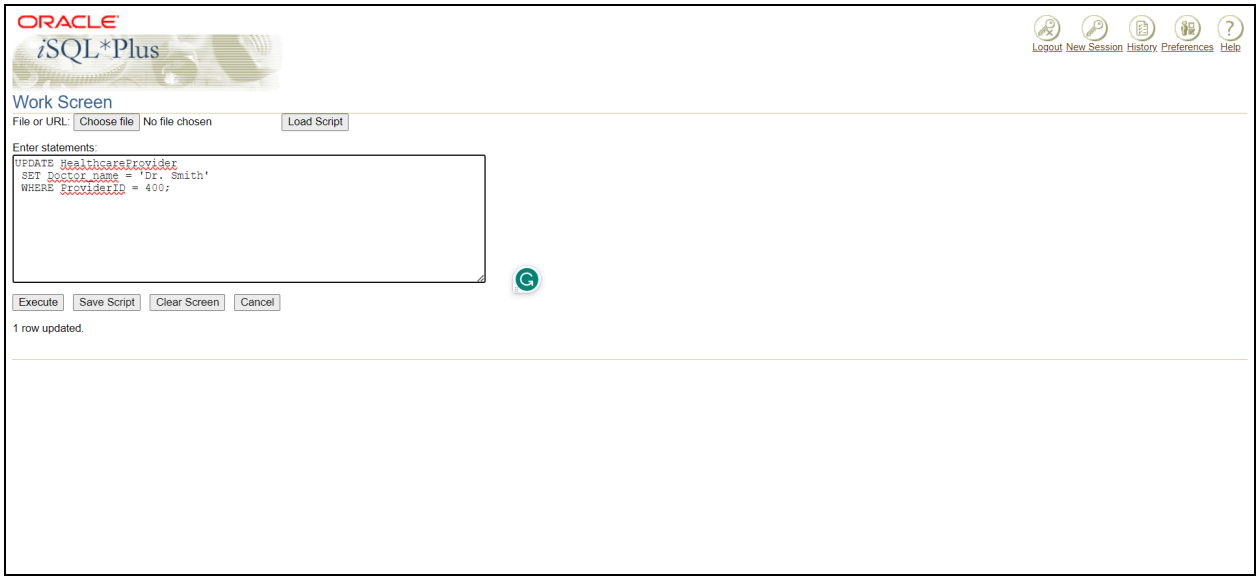
```
- - Group by vaccine quality, more than 1 occurrence
SELECT Quality, COUNT(Quality)
    FROM InventoryCenter
      GROUP BY Quality
      HAVING COUNT(Quality) > 1;
```

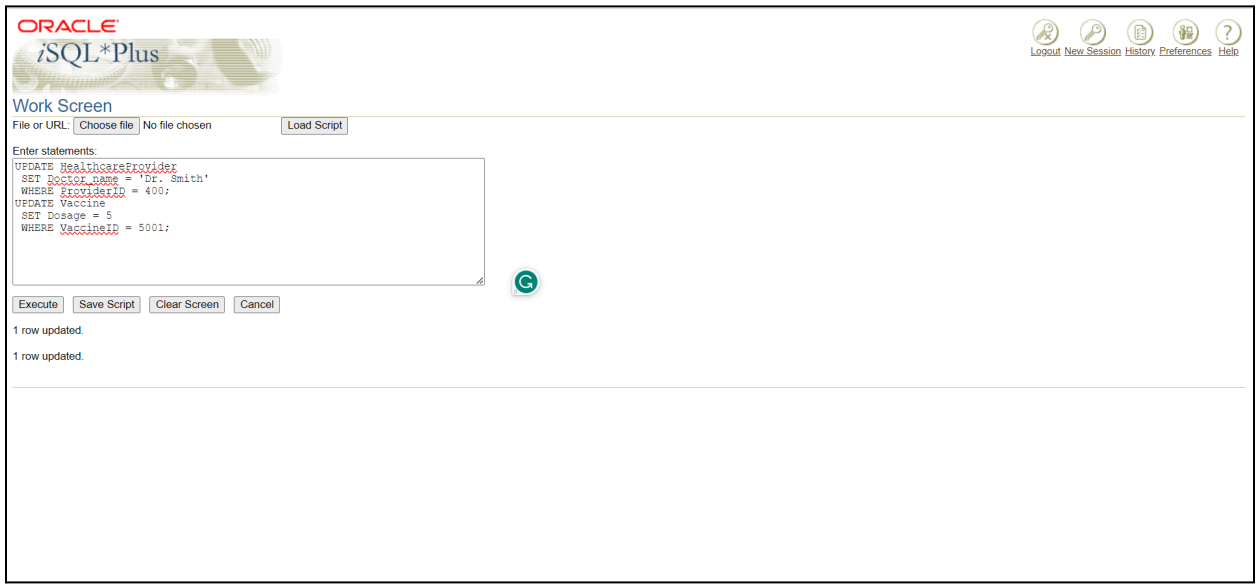


This query will count the number of occurrences of good quality or bad quality. It will display if the number is greater than 1. Since we have 3 that are ‘good’ and 1 that is ‘bad’ it will only display the good quality.

```
- Update Statement one
UPDATE HealthcareProvider
  SET Doctor_name = 'Dr. Smith'
  WHERE ProviderID = 400;
- Update Statement two
UPDATE Vaccine
  SET Dosage = 5
  WHERE VaccineID = 5001;
```

The first statement updates the ‘Doctor_name’ to ‘Dr.Smith’ for the healthcare provider with ‘ProviderID’ 400 in the ‘healthcareProvider’ table.
The second statement updates the dosage of the vaccine with ‘VaccineID’ 5001 to 5 in the ‘Vaccine’ table.





Business situation:

1) We are a team tasked with developing a comprehensive database system for a healthcare organization to manage a COVID-19 vaccination program.
2) The database will include following business functions/situations:
   Tracking vaccine distribution, Records of the patients, Inventory management, Appointment information, Vaccination data.

Business rules:

1) Appointment Scheduling: Based on vaccine availability and eligibility standards established by public health recommendations, patients can plan immunization appointments. By giving those who fit certain requirements, such as age or underlying health concerns, priority, this guideline makes sure that appointments are scheduled in an

orderly fashion.

2) Management of vaccine inventory: Vaccine doses that are getting close to expiration should be used first. This regulation guarantees that vaccines are provided before they expire and reduces vaccine waste.
3) Reporting: Regular reports on the distribution of vaccines, demographic information, and any unfavorable events must be produced for public health authorities. Transparency, accountability, and adherence to reporting requirements are all made easier by this rule.
4) Patient Consent: Prior to getting the vaccine, patients must give their informed consent, and consent records must be kept in a secure location. The ethical and legal requirements for patient autonomy and data privacy are upheld by this rule.
5) Vaccination Records: Each vaccination must have a precise record that includes the type of vaccine, the lot number, the dose, and patient information. In order to monitor vaccine effectiveness and safety, this rule guarantees the integrity of immunization data.
6) Distribution of Vaccines: Healthcare professionals must follow regulations for the distribution of vaccines that provide priority to high-risk populations and regions. This regulation makes sure that vaccines are distributed effectively and fairly to those who need them the most.

A. Business Functions :

1) Scheduling an appointment: Depending on the region and vaccine supply, patients and medical professionals can book vaccination appointments.
2) Vaccine Inventory Management: Monitoring vaccine doses, shelf lives, and delivery to multiple locations.
3) Reporting: Producing statistics on the distribution of vaccines, patient demographics, and adverse events for public health authorities, medical professionals, and administrators.
4) Patient registration: It involves gathering and keeping track of patient data, such as consent papers and medical histories.
5) Vaccination Records: Keeping track of and updating patient immunization records, including doses given and any negative effects.
6) Information on the healthcare provider: descriptions of the healthcare workers' specialties and access rights.
7) Administrative Functions: Access control, system configuration, and user administration.

B. Sample Data stored -

1) Patient Data - Full name, DOB, Contact information, Medical records, etc.
2) Vaccine Data - Type of the vaccine, Manufacturer, etc.
3) Appointment Data - Appointment date & time, Clinic location, Staff member assigned, etc.
4) Inventory Data - Vaccine stock levels, shipments received, Supply forecasts, etc.
5) Logistical Data - records of the vaccine distribution, Shipping and delivery information,

etc.

6) Feedback & Survey Data - Feedback and satisfaction survey responses of the patients, improvement suggestions.

Assumptions:

1) We will prioritize patients who are immunocompromised.
2) We will assume patients are willing to wait a few weeks to be vaccinated.
3) We are trying to assure as many people as possible are able to be vaccinated.
4) We will target patients by advertising.
5) We will assume patients have the resources to be able to get to their vaccination appointment.
6) We will assume that we have no previous records of patients and how many doses of vaccines they already have.
7) We will assume vaccines are free of charge so no financial data needed.

References:

https://www.cdc.gov/vaccines/covid-19/planning/considerations-operating-vaccine-clinic.html
https://www.oracle.com/news/announcement/oracle-cloud-manages-covid-19-vaccination-program-121520/
https://www.who.int/news/item/17-05-2022-statement-for-healthcare-professionals-how-covid-19-vaccines-are-regulated-for-safety-and-effectiveness

Project Deliverable - II

Entities:

1) Vaccine
2) Patient
3) Distribution Center
4) Inventory
5) Appointment Record
6) Healthcare Provider

Relationships:

- Patient **Schedules** Appointment: Mandatory one to Many Mandatory - A patient must have at least one appointment record and can have multiple. An appointment record must be attributed to at least one patient and at most one patient.
- Patient **Receives** Vaccine : Mandatory one to optional many from vaccine to patient - One patient can receive mandatory one vaccine to multiple booster vaccines.

- ❖ Distribution Center **Manages** Inventory: Mandatory one to optional many from Distribution Center to Inventory - A distribution center allocates and manages one or many inventories.
- ❖ Healthcare Provider **Administers** Vaccine: Mandatory one to optional many from Healthcare Provider to Vaccine - One healthcare provider is required to administer at least one vaccine.
- ❖ Patient **Assigned to** Healthcare provider: Mandatory many to mandatory one from Patient to Healthcare provider - Many patients will be assigned to one healthcare provider. One healthcare provider can have multiple patients, but one patient cannot have multiple healthcare providers.
- ❖ Distribution Center **Distributes** Vaccine: Mandatory one to optional many from Distribution center to vaccine - One distribution center will distribute at least one vaccine.
- ❖ Distribution Center **Belongs to** Healthcare provider: Mandatory one to optional many from Distribution Center to Healthcare provider - One distribution center belongs to at least one healthcare provider.

Attributes:

1) Vaccine attributes: VaccineID, Name, Manufacturer, Type_of_vaccine, Dosage. VaccineID is the identifier for the particular vaccine, Name is the vaccine name, Manufacturer is the manufacturer of the vaccine, Type_of_vaccine is the vaccine type, and Dosage is the vaccine dosage.
2) Patient attributes: PatientID, Name, Contact_info, Immunocompromised_status, Patient_records. PatientID is the patient identifier, Name is the patient's name, Contact_info is the patient's contact info, Immunocompromised_status is whether or not the patient is immunocompromised, and Patient_records is the records on file for the patient.
3) Distribution Center attributes: CenterID, Location, Capacity. CenterID is the identifier of the distribution center, Location is the location of the distribution center, and Capacity is how much inventory the distribution center can hold.
4) Inventory attributes: InventoryID, Quantity, Expiry_date, Stock_levels, Supply_forecast, Shippment_status. InventoryID is the identifier of the inventory, Quantity is the amount of inventory available, Expiry_date is the date the inventory will expire, Stock_levels is the level of stock of the inventory, Supply_forecast is the projected supply, and Shippment_status is the status of the shipment of inventory.
5) Appointment attributes: AppointmentID, Date, Time, Location. AppointmentID is the identifier for the appointment, Date is the date of the appointment, Time is the appointment time, and Location is the appointment location.
6) Healthcare Provider attributes: ProviderID, DoctorID, Doctor_name, InsuranceID, Specialty. ProviderID is the identifier of the healthcare provider, DoctorID is the

identifier of the doctor, Doctor_name is the name of the doctor, InsuranceID is the identifier of the insurance, and Specialty is the specialty of the healthcare provider.

Description for each entity:

Vaccine:

- VaccineID (Key Identifier) - Required, Simple, Single-Valued
- Name - Required, Simple, Single-Valued
- Manufacturer - Required, Simple, Multi-Valued
- Type_of_vaccine - Required, Simple, Single-Valued
- Dosage - Required, Simple, Single-Valued

Patient:

- PatientID (Key Identifier) - Required, Simple, Single-Valued
- Name - Required, Simple, Single-Valued
- Contact_info - Required, Composite (may include multiple contact details), Multi-Valued (e.g., multiple phone numbers or addresses)
- Immunocompromised_status - Optional, Simple, Single-Valued
- Patient_records - Optional, Simple, Single-Valued

Distribution Center:

- CenterID (Key Identifier) - Required, Simple, Single-Valued
- Location - Required, Composite, Single-Valued
- Capacity - Required, Simple, Single-Valued

Inventory:

- InventoryID (Key Identifier) - Required, Simple, Single-Valued
- Quantity - Required, Simple, Single-Valued
- Expiry_date - Required, Simple, Single-Valued
- Stock_levels - Required, Simple, Single-Valued
- Supply_forecast - Required, Simple, Single-Valued
- Shippment_status - Required, Simple, Single-Valued

Appointment:

- AppointmentID (Key Identifier) - Required, Simple, Single-Valued
- Date - Required, Simple, Single-Valued
- Time - Required, Simple, Single-Valued
- Location - Required, composite, Single-Valued

Healthcare Provider:

- ProviderID (Key Identifier) - Required, Simple, Single-Valued
- DoctorID - Required, Simple, Single-Valued
- Doctor_name - Required, Simple, Single-Valued
- InsuranceID - Required, Simple, Single-Valued
- Specialty - Required, Simple, Single-Valued

Project Deliverable - III

The "Appointment" and "Patient" tables are established as having a one-to-many association. A clear relationship between appointment records and associated patients is ensured in this scenario by using the main key on the "Appointment" side as the foreign key on the "Patient" side.
In the same way, the main key in the "Healthcare Provider" database acts as a foreign key in the "Vaccine," "Patient," and "Distribution Center" tables. One healthcare provider may be connected to several patients, vaccinations, or distribution facilities. This configuration denotes one-to-many relationships.
A one-to-many relationship between distribution centers and vaccinations is highlighted by the primary key in the "Distribution Center" database serving as a foreign key in the "Vaccine" table, further expanding this relational structure.
A many-to-many relationship is identified for the "Inventory" and "Distribution Center" tables. A new relation called "Management" is established in order to describe this complex association in an effective manner. The many-to-many link between inventory items and distribution centers is

seamlessly captured by this intermediary table, which has the primary keys of "Inventory" and "Distribution Center" as its composite primary keys. This makes sure that the database structure is normalized and ordered, which facilitates effective data administration and retrieval procedures.

Project Deliverable - IV

Healthcare Provider Table:

ProviderID	DoctorID	Doctor_name	InsuranceID	Specialty
400	6000	Perry	70000	Pediatrics
401	6001	Figueroa	70001	Oncology
402	6002	Russell	70002	Pulmonology
403	6003	Santos	70003	Infectious diseases

Vaccine Table:

VaccineID	Name	Manufacturer	Dosage	Type_of_vaccine
5000	Spike Vax	Moderna	3	mRNA
5001	Comirnaty	Pfizer	4	mRNA
5002	Covishield	AstraZeneca	1	vector
5003	Nuvaxovid	Novavax	2	protein subunit

Distribution Center Table:

CenterID	ProviderID	Location	Capacity
3000	401	Baltimore	100000
3001	403	Colombia	50000
3002	400	Salisbury	60000
3003	401	Catonsville	80000

Appointment Record:

AppointmentID	Date	Time	Location	PatientID
7000	12/6/21	4.48pm	Baltimore	802
7001	9/8/22	11.03am	Colombia	801
7002	3/9/21	10.22am	Salisbury	800
7003	15/12/22	12.30pm	Catonsville	804

Patient Record Table:

PaitentID	Name	Contact Info	Immunocompromised_status	Patient_record	ProviderID

800	Peter	698-587-963	Yes	B01	400
801	Andrea	459-863-951	No	B02	401
802	Monica	785-951-357	No	B03	402
803	Brian	698-357-951	Yes	B04	403

Inventory Center Table:

InventoryID	Quality	Expiry_date	CenterID	VaccineID
1100	Good	15/1/25	3000	5000
1101	Bad	29/6/24	3001	5001
1102	Good	19/8/25	3002	5002
1103	Good	23/3/24	3003	5003

Management Center:

InventoryID	CenterID
1100	3000
1101	3001
1102	3002
1103	3003

SQL statements for the application:

```
CREATE TABLE HealthcareProvider (
    ProviderID INT PRIMARY KEY,
    DoctorID INT,
```

```
    Doctor_name VARCHAR(255),
    InsuranceID INT,
    Specialty VARCHAR(255)
);
INSERT INTO HealthcareProvider (ProviderID, DoctorID, Doctor_name, InsuranceID, Specialty)
VALUES (400, 6000, 'Perry', 70000, 'Pediatrics');
INSERT INTO HealthcareProvider (ProviderID, DoctorID, Doctor_name, InsuranceID, Specialty)
VALUES (401, 6001, 'Figueroa', 70001, 'Oncology');
INSERT INTO HealthcareProvider (ProviderID, DoctorID, Doctor_name, InsuranceID, Specialty)
VALUES (402, 6002, 'Russel', 70002, 'Pulmonology');
INSERT INTO HealthcareProvider (ProviderID, DoctorID, Doctor_name, InsuranceID, Specialty)
VALUES (403, 6003, 'Santos', 70003, 'Infectious diseases’);

-- Create the Vaccine table
CREATE TABLE Vaccine (
    VaccineID INT PRIMARY KEY,
    Name VARCHAR(255),
    Manufacturer VARCHAR(255),
    Dosage INT,
    Type_of_vaccine VARCHAR(255)
);

-- Insert data into the Vaccine table
INSERT INTO Vaccine (VaccineID, Name, Manufacturer, Dosage, Type_of_vaccine)
VALUES (5000, 'Spike Vax', 'Moderna', 3, 'mRNA');
INSERT INTO Vaccine (VaccineID, Name, Manufacturer, Dosage, Type_of_vaccine)
VALUES (5001, 'Comirnaty', 'Pfizer', 4, 'mRNA');
INSERT INTO Vaccine (VaccineID, Name, Manufacturer, Dosage, Type_of_vaccine)
VALUES (5002, 'Covishield', 'AstraZeneca', 1, 'Vector');
INSERT INTO Vaccine (VaccineID, Name, Manufacturer, Dosage, Type_of_vaccine)
VALUES (5003, 'Novaxovid', 'Novavax', 2, 'Protein subunit');

-- Create the DistributionCenter table
CREATE TABLE DistributionCenter (
    CenterID INT PRIMARY KEY,
    ProviderID INT,
    Location VARCHAR(255),
    Capacity INT,
    FOREIGN KEY (ProviderID) REFERENCES HealthcareProvider(ProviderID)
);

-- Insert data into the DistributionCenter table
INSERT INTO DistributionCenter (CenterID, ProviderID, Location, Capacity)
VALUES (3000, 401, 'Baltimore', 100000);
INSERT INTO DistributionCenter (CenterID, ProviderID, Location, Capacity)
VALUES (3001, 403, 'Colombia', 50000);
INSERT INTO DistributionCenter (CenterID, ProviderID, Location, Capacity)
VALUES (3002, 400, 'Salisbury', 60000);
INSERT INTO DistributionCenter (CenterID, ProviderID, Location, Capacity)
VALUES (3003, 401, 'Catonsville', 80000);

-- Create the PatientRecord table
CREATE TABLE PatientRecord (
    PatientID INT PRIMARY KEY,
    Name VARCHAR(255),
    ContactInfo VARCHAR(255),
    Immunocompromised_status VARCHAR(3),
    Patient_record VARCHAR(10),
    ProviderID INT,
    FOREIGN KEY (ProviderID) REFERENCES HealthcareProvider(ProviderID)
);

-- Create the AppointmentRecord table
CREATE TABLE AppointmentRecord (
    AppointmentID INT PRIMARY KEY,
    "Date" DATE,
```

```
    "Time" TIMESTAMP,
    Location VARCHAR(255),
    PatientID INT,
    FOREIGN KEY (PatientID) REFERENCES PatientRecord(PatientID)
);

-- Insert data into the PatientRecord table
INSERT INTO PatientRecord (PatientID, Name, ContactInfo, Immunocompromised_status, Patient_record,
ProviderID)
VALUES (800, 'Peter', '698-587-963', 'Yes', 'B01', 400);
INSERT INTO PatientRecord (PatientID, Name, ContactInfo, Immunocompromised_status, Patient_record,
ProviderID)
VALUES (801, 'Andrea', '459-863-951', 'No', 'B02', 401);
INSERT INTO PatientRecord (PatientID, Name, ContactInfo, Immunocompromised_status, Patient_record,
ProviderID)
VALUES (802, 'Monica', '785-951-357', 'No', 'B03', 402);
INSERT INTO PatientRecord (PatientID, Name, ContactInfo, Immunocompromised_status, Patient_record,
ProviderID)
VALUES (803, 'Brian', '698-357-951', 'Yes', 'B04', 403);

-- Insert data into the AppointmentRecord table
INSERT INTO AppointmentRecord (AppointmentID, "Date", "Time", Location, PatientID)
VALUES (7000, TO_DATE('2021-12-06', 'YYYY-MM-DD'), TO_TIMESTAMP('2021-12-06 16:48:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Baltimore', 802);
INSERT INTO AppointmentRecord (AppointmentID, "Date", "Time", Location, PatientID)
VALUES (7001, TO_DATE('2022-09-08', 'YYYY-MM-DD'), TO_TIMESTAMP('2022-09-08 11:03:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Colombia', 801);
INSERT INTO AppointmentRecord (AppointmentID, "Date", "Time", Location, PatientID)
VALUES (7002, TO_DATE('2021-03-09', 'YYYY-MM-DD'), TO_TIMESTAMP('2021-03-09 10:22:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Salisbury', 800);
INSERT INTO AppointmentRecord (AppointmentID, "Date", "Time", Location, PatientID)
VALUES (7003, TO_DATE('2022-12-15', 'YYYY-MM-DD'), TO_TIMESTAMP('2022-12-15 12:30:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Catonsville', 803);

-- Create the InventoryCenter table
CREATE TABLE InventoryCenter (
    InventoryID INT PRIMARY KEY,
    Quality VARCHAR(255),
    Expiry_date DATE,
    CenterID INT,
    VaccineID INT,
    FOREIGN KEY (CenterID) REFERENCES DistributionCenter(CenterID),
    FOREIGN KEY (VaccineID) REFERENCES Vaccine(VaccineID)
);

-- Create the ManagementCenter table
CREATE TABLE ManagementCenter (
    InventoryID INT PRIMARY KEY,
    CenterID INT,
    FOREIGN KEY (InventoryID) REFERENCES InventoryCenter(InventoryID),
    FOREIGN KEY (CenterID) REFERENCES DistributionCenter(CenterID)
);
-- Insert data into the InventoryCenter table
INSERT INTO InventoryCenter (InventoryID, Quality, Expiry_date, CenterID, VaccineID)
VALUES (1100, 'Good', TO_DATE('2025-01-15', 'YYYY-MM-DD'), 3000, 5000);
INSERT INTO InventoryCenter (InventoryID, Quality, Expiry_date, CenterID, VaccineID)
VALUES (1101, 'Bad', TO_DATE('2024-06-29', 'YYYY-MM-DD'), 3001, 5001);
INSERT INTO InventoryCenter (InventoryID, Quality, Expiry_date, CenterID, VaccineID)
VALUES (1102, 'Good', TO_DATE('2025-08-19', 'YYYY-MM-DD'), 3002, 5002);
INSERT INTO InventoryCenter (InventoryID, Quality, Expiry_date, CenterID, VaccineID)
VALUES (1103, 'Good', TO_DATE('2024-03-23', 'YYYY-MM-DD'), 3003, 5003);

-- Insert data into the ManagementCenter table
INSERT INTO ManagementCenter (InventoryID, CenterID)
VALUES (1100, 3000);
INSERT INTO ManagementCenter (InventoryID, CenterID)
```

```
VALUES (1101, 3001);
INSERT INTO ManagementCenter (InventoryID, CenterID)
VALUES (1102, 3002);
INSERT INTO ManagementCenter (InventoryID, CenterID)
VALUES (1103, 3003);



—Join Statement
SELECT
    mc.InventoryID,
    ic.Quality,
    ic.Expiry_date,
    ic.CenterID AS InventoryCenter_CenterID,
    dc.Location AS InventoryCenter_Location,
    ic.VaccineID,
    vc.Name AS Vaccine_Name,
    mc.CenterID AS ManagementCenter_CenterID,
    dc.Location AS ManagementCenter_Location
FROM
    ManagementCenter mc
JOIN
    InventoryCenter ic ON mc.InventoryID = ic.InventoryID
JOIN
    DistributionCenter dc ON ic.CenterID = dc.CenterID
JOIN
    Vaccine vc ON ic.VaccineID = vc.VaccineID;
```

A set of rows with data from the ManagementCenter, InventoryCenter, DistributionCenter, and Vaccine databases linked together in accordance with the predetermined criteria will be the end result.This query joins linked tables in a database in order to extract specific information. It provides a thorough view of inventory and management information by combining data from several tables based on shared columns.