Dataware house assignment-1

1. **Design a Data Warehouse for IPL Cricket Tournament**:

Fact Table: Matches
match_id (primary key)
match_date
team_1_id (foreign key to Teams dimension)
team_2_id (foreign key to Teams dimension)
venue_id (foreign key to Venues dimension)
winner_id (foreign key to Teams dimension)
Dimension Tables:

Teams
team_id (primary key)
team_name
team_city
Players
player_id (primary key)
player_name
team_id (foreign key to Teams dimension)
player_role (batsman, bowler, wicketkeeper, etc.)
Venues
venue_id (primary key)
venue_name
venue_city
SQL Queries:

Total matches played by each team:
SELECT team_name, COUNT(*) FROM Matches JOIN Teams ON Matches.team_1_id = Teams.team_id OR Matches.team_2_id = Teams.team_id GROUP BY team_name;
Average runs scored by each team:
SELECT team_name, AVG(runs_scored) FROM Matches JOIN Teams ON Matches.team_1_id = Teams.team_id OR Matches.team_2_id = Teams.team_id JOIN Innings ON Matches.match_id = Innings.match_id GROUP BY team_name;

2. **Design a Data Warehouse for Food delivery app like Swiggy, Zomato:**

Fact Table: Orders
order_id (primary key)
order_date
customer_id (foreign key to Customers dimension)
restaurant_id (foreign key to Restaurants dimension)
delivery_address_id (foreign key to Addresses dimension)
total_cost

Dimension Tables:

Customers
customer_id (primary key)
customer_name
customer_email
customer_phone
Restaurants
restaurant_id (primary key)
restaurant_name
restaurant_city
restaurant_category (veg, non-veg, etc.)
Addresses
address_id (primary key)
address_line_1
address_line_2
city
state
zip
country
Payment
payment_id (primary key)
order_id (foreign key to Orders)
payment_amount
payment_method (cash, card, digital wallet, etc.)
payment_status (success, failure)
SQL Queries:

Total sales by each restaurant:
SELECT restaurant_name, SUM(total_cost) FROM Orders JOIN Restaurants ON Orders.restaurant_id = Restaurants.restaurant_id GROUP BY restaurant_name;
Total orders by each customer:
SELECT customer_name, COUNT(*) FROM Orders JOIN Customers ON Orders.customer_id = Customers.customer_id GROUP BY customer_name;
Total sales by payment method:
SELECT payment_method, SUM(payment_amount) FROM Payment GROUP BY payment_method;

3. **Design a Data Warehouse for cab ride service like Uber, Lyft:**

Fact Table: Rides
ride_id (primary key)
ride_start_time
ride_end_time
driver_id (foreign key to Drivers dimension)
rider_id (foreign key to Riders dimension)
start_location_id (foreign key to Locations dimension)

end_location_id (foreign key to Locations dimension)
total_cost
Dimension Tables:

Drivers
driver_id (primary key)
driver_name
driver_phone
driver_email
driver_rating
Riders
rider_id (primary key)
rider_name
rider_phone
rider_email
Locations
location_id (primary key)
location_name
city
state
country
Vehicles
vehicle_id (primary key)
vehicle_make
vehicle_model
vehicle_year
driver_id (foreign key to Drivers dimension)
SQL Queries:

Total rides by each driver:
SELECT driver_name, COUNT(*) FROM Rides JOIN Drivers ON Rides.driver_id = Drivers.driver_id GROUP BY driver_name;
Total rides by each rider:
SELECT rider_name, COUNT(*) FROM Rides JOIN Riders ON Rides.rider_id = Riders.rider_id GROUP BY rider_name;
Total revenue by vehicle make:
SELECT vehicle_make, SUM(total_cost) FROM Rides JO

4. **Design a Data Warehouse for Restaurent table booking app like Dineout:**

Fact Table: Bookings
booking_id (primary key)
booking_date
customer_id (foreign key to Customers dimension)
restaurant_id (foreign key to Restaurants dimension)

table_id (foreign key to Tables dimension)
booking_status (confirmed, cancelled)
Dimension Tables:

Customers
customer_id (primary key)
customer_name
customer_email
customer_phone
Restaurants
restaurant_id (primary key)
restaurant_name
restaurant_city
restaurant_category (veg, non-veg, etc.)
Tables
table_id (primary key)
table_number
table_capacity
restaurant_id (foreign key to Restaurants dimension)
Time slots
time_slot_id (primary key)
time_slot_start
time_slot_end
restaurant_id (foreign key to Restaurants dimension)
SQL Queries:

Total bookings by each restaurant:
SELECT restaurant_name, COUNT(*) FROM Bookings JOIN Restaurants ON Bookings.restaurant_id = Restaurants.restaurant_id GROUP BY restaurant_name;
Total bookings by each customer:
SELECT customer_name, COUNT(*) FROM Bookings JOIN Customers ON Bookings.customer_id = Customers.customer_id GROUP BY customer_name;
Total cancelled bookings by each restaurant:
SELECT restaurant_name, COUNT(*) FROM Bookings JOIN Restaurants ON Bookings.restaurant_id = Restaurants.restaurant_id WHERE booking_status = 'cancelled' GROUP BY restaurant_name;

5. Design a Data Warehouse for Covid Vaccination Application:

Fact Table: Vaccinations
vaccination_id (primary key)
vaccination_date
patient_id (foreign key to Patients dimension)
vaccination_center_id (foreign key to Vaccination Centers dimension)
vaccination_status (completed, pending)
Dimension Tables:

Patients
patient_id (primary key)
patient_name
patient_age
patient_gender
patient_address
Vaccination Centers
vaccination_center_id (primary key)
center_name
center_address
center_city
center_state
Vaccines
vaccine_id (primary key)
vaccine_name
vaccine_manufacturer
vaccine_batch
Staff
staff_id (primary key)
staff_name
staff_role (nurse, doctor, administrator)
vaccination_center_id (foreign key to Vaccination Centers dimension)
SQL Queries:

Total vaccinations by each center:
SELECT center_name, COUNT(*) FROM Vaccinations JOIN Vaccination_Centers ON Vaccinations.vaccination_center_id = Vaccination_Centers.vaccination_center_id GROUP BY center_name;
Total vaccinations by each staff:
SELECT staff_name, COUNT(*) FROM Vaccinations JOIN Staff ON Vaccinations.staff_id = Staff.staff_id GROUP BY staff_name;