**Design Data Warehouses for Given Below Products:**

**Note:** While designing any Data Warehouse make sure to cover given below points.

a. Design Fact & Dimension tables

b. Create meaningful Primary & Foreign keys

c. Try to follow Star/Snowflake Schema Design

d. Try to write few SQL queries to generate insightful business metrics (This is the critical

point because you need to understand the Data & Business both)

**Q1. Design a Data Warehouse for IPL Cricket Tournament (Asked in Flipkart Interview for**

**Senior Data Engineer role)**

**ANSWER:**

Designing a Data Warehouse for IPL Cricket Tournament involves gathering data about various aspects of the tournament such as matches, players, teams, venues, and various statistics related to the matches. Here is a possible design for the Data Warehouse:

**Fact Tables:**

a. Match Fact Table:

* Match\_ID (PK)
* Date
* Venue\_ID (FK)
* Toss\_Winner\_ID (FK)
* Toss\_Decision
* Match\_Winner\_ID (FK)
* Win\_Type
* Player\_of\_Match\_ID (FK)
* Umpire\_1\_ID (FK)
* Umpire\_2\_ID (FK)
* Umpire\_3\_ID (FK)
* Winning\_Margin
* Is\_Super\_Over

b. Batting Fact Table:

* Match\_ID (PK, FK)
* Player\_ID (PK, FK)
* Runs
* Balls
* Fours
* Sixes
* Strike\_Rate
* Dismissal\_Type
* Dismissal\_By

c. Bowling Fact Table:

* Match\_ID (PK, FK)
* Player\_ID (PK, FK)
* Overs
* Runs
* Wickets
* Economy
* Maiden\_Overs
* No\_Balls
* Wide\_Balls

**Dimension Tables:**

a. Match Dimension Table:

* Match\_ID (PK)
* Date
* Venue\_ID (FK)
* Team\_1\_ID (FK)
* Team\_2\_ID (FK)
* Season\_ID (FK)

b. Venue Dimension Table:

* Venue\_ID (PK)
* Venue\_Name
* City
* State
* Country

c. Season Dimension Table:

* Season\_ID (PK)
* Season\_Year

d. Team Dimension Table:

* Team\_ID (PK)
* Team\_Name

e. Player Dimension Table:

* Player\_ID (PK)
* Player\_Name
* DOB
* Batting\_Hand
* Bowling\_Skill
* Country

f. Umpire Dimension Table:

* Umpire\_ID (PK)
* Umpire\_Name

**Primary & Foreign Keys:**

* Match\_ID is the Primary Key of Match Fact Table and Foreign Key in all other fact tables.
* Venue\_ID, Team\_ID, Player\_ID, Season\_ID, and Umpire\_ID are Primary Keys in the respective Dimension tables, and Foreign Keys in the Fact Tables.

**Star Schema Design:**

* The Fact Tables (Match, Batting, and Bowling) are connected to Dimension Tables (Match, Venue, Season, Team, Player, and Umpire) through the Primary and Foreign keys.
* The design follows the Star Schema Design, with the Fact Tables forming the centre of the schema and the Dimension Tables forming the surrounding points.

**Sample SQL Queries:**

a. Count the number of matches played in each season:

SELECT season\_dimension.Season\_Year, COUNT(match\_dimension.Match\_ID) AS Matches\_Played

FROM match\_fact\_table

INNER JOIN season\_dimension ON match\_dimension.Season\_ID = season\_dimension.Season\_ID

GROUP BY season\_dimension.Season\_Year;

b. Calculate the total runs scored by each player in a particular season:

SELECT player\_dimension.Player\_Name, SUM(batting\_fact\_table.Runs) AS Total\_Runs\_Scored

FROM batting\_fact\_table

INNER JOIN match\_dimension ON batting\_fact\_table.Match\_ID = match\_dimension.Match\_ID

INNER JOIN player\_dimension ON batting\_fact\_table.Player\_ID

**Q2. Design a Data Warehouse for Food delivery app like Swiggy, Zomato (Asked in Grab**

**for Data Engineer role)**

**ANSWER:**

Designing a Data Warehouse for a food delivery app like Swiggy or Zomato involves gathering data about various aspects of the app such as customers, restaurants, orders, delivery partners, and various statistics related to these entities. Here is a possible design for the Data Warehouse:

**Fact Tables:**

a. Order Fact Table:

* Order\_ID (PK)
* Customer\_ID (FK)
* Restaurant\_ID (FK)
* Delivery\_Partner\_ID (FK)
* Order\_Status
* Payment\_Mode
* Order\_Amount
* Order\_Date
* Order\_Time
* Delivery\_Date
* Delivery\_Time
* Delivery\_Address
* Discount\_Amount

b. Payment Fact Table:

* Payment\_ID (PK)
* Order\_ID (FK)
* Payment\_Amount
* Payment\_Date
* Payment\_Time
* Payment\_Status

**Dimension Tables:**

a. Customer Dimension Table:

* Customer\_ID (PK)
* Customer\_Name
* Customer\_Email
* Customer\_Phone
* Customer\_Address

b. Restaurant Dimension Table:

* Restaurant\_ID (PK)
* Restaurant\_Name
* Restaurant\_Address
* Restaurant\_Phone
* Restaurant\_Cuisine

c. Delivery Partner Dimension Table:

* Delivery\_Partner\_ID (PK)
* Partner\_Name
* Partner\_Phone
* Partner\_Vehicle\_Number
* Partner\_Vehicle\_Type

d. Payment Mode Dimension Table:

* Payment\_Mode\_ID (PK)
* Payment\_Mode\_Name

e. Order Status Dimension Table:

* Order\_Status\_ID (PK)
* Order\_Status\_Name

**Primary & Foreign Keys:**

* Order\_ID is the Primary Key of Order Fact Table and Foreign Key in Payment Fact Table.
* Customer\_ID, Restaurant\_ID, and Delivery\_Partner\_ID are Foreign Keys in the Order Fact Table, and Primary Keys in the respective Dimension Tables.
* Payment\_Mode\_ID and Order\_Status\_ID are Foreign Keys in the Order Fact Table, and Primary Keys in the respective Dimension Tables.

**Star Schema Design:**

* The Fact Tables (Order and Payment) are connected to Dimension Tables (Customer, Restaurant, Delivery Partner, Payment Mode, and Order Status) through the Primary and Foreign keys.
* The design follows the Star Schema Design, with the Fact Tables forming the centre of the schema and the Dimension Tables forming the surrounding points.

**Sample SQL Queries:**

a. Calculate the total amount earned by each restaurant in a particular month:

SELECT restaurant\_dimension.Restaurant\_Name, SUM(order\_fact\_table.Order\_Amount) AS Total\_Amount\_Earned

FROM order\_fact\_table

INNER JOIN restaurant\_dimension ON order\_fact\_table.Restaurant\_ID = restaurant\_dimension.Restaurant\_ID

WHERE MONTH(order\_fact\_table.Order\_Date) = '3' AND YEAR(order\_fact\_table.Order\_Date) = '2023'

GROUP BY restaurant\_dimension.Restaurant\_Name;

b. Count the number of orders delivered by each delivery partner in a particular week:

SELECT delivery\_partner\_dimension.Partner\_Name, COUNT(order\_fact\_table.Order\_ID) AS Orders\_Delivered

FROM order\_fact\_table

INNER JOIN delivery\_partner\_dimension ON order\_fact\_table.Delivery\_Partner\_ID = delivery\_partner\_dimension.Delivery\_Partner\_ID

WHERE WEEK(order\_fact\_table.Delivery\_Date) = '13' AND YEAR(order\_fact\_table.Delivery\_Date) = '2023'

GROUP BY delivery\_partner\_dimension.Partner\_Name;

**Q3. Design a Data Warehouse for cab ride service like Uber, Lyft (Asked in Google for Data**

**Engineer role)**

**ANSWER:**

Designing a Data Warehouse for a cab ride service like Uber or Lyft involves collecting data about the trips, drivers, vehicles, riders, and various metrics related to these entities. Here is a possible design for the Data Warehouse:

**Fact Tables:**

a. Trip Fact Table:

* Trip\_ID (PK)
* Rider\_ID (FK)
* Driver\_ID (FK)
* Vehicle\_ID (FK)
* Pickup\_Location\_ID (FK)
* Drop\_Location\_ID (FK)
* Trip\_Date
* Trip\_Time
* Trip\_Status
* Fare\_Amount
* Surge\_Multiplier

b. Payment Fact Table:

* Payment\_ID (PK)
* Trip\_ID (FK)
* Payment\_Amount
* Payment\_Date
* Payment\_Time
* Payment\_Method

**Dimension Tables:**

a. Rider Dimension Table:

* Rider\_ID (PK)
* Rider\_Name
* Rider\_Email
* Rider\_Phone
* Rider\_Address

b. Driver Dimension Table:

* Driver\_ID (PK)
* Driver\_Name
* Driver\_Email
* Driver\_Phone
* Driver\_Address

c. Vehicle Dimension Table:

* Vehicle\_ID (PK)
* Vehicle\_Type
* Vehicle\_Model
* Vehicle\_Registration\_Number

d. Location Dimension Table:

* Location\_ID (PK)
* Location\_Name
* Location\_Latitude
* Location\_Longitude

e. Payment Method Dimension Table:

* Payment\_Method\_ID (PK)
* Payment\_Method\_Name

f. Trip Status Dimension Table:

* Trip\_Status\_ID (PK)
* Trip\_Status\_Name

**Primary & Foreign Keys:**

* Trip\_ID is the Primary Key of Trip Fact Table and Foreign Key in Payment Fact Table.
* Rider\_ID, Driver\_ID, Vehicle\_ID, Pickup\_Location\_ID, and Drop\_Location\_ID are Foreign Keys in the Trip Fact Table and Primary Keys in the respective Dimension Tables.
* Payment\_Method\_ID and Trip\_Status\_ID are Foreign Keys in the Trip Fact Table and Primary Keys in the respective Dimension Tables.

**Star Schema Design:**

* The Fact Tables (Trip and Payment) are connected to Dimension Tables (Rider, Driver, Vehicle, Location, Payment Method, and Trip Status) through the Primary and Foreign keys.
* The design follows the Star Schema Design, with the Fact Tables forming the center of the schema and the Dimension Tables forming the surrounding points.

**Sample SQL Queries:**

a. Calculate the total number of trips taken by each driver in a particular month:

SELECT driver\_dimension.Driver\_Name, COUNT(trip\_fact\_table.Trip\_ID) AS Total\_Trips

FROM trip\_fact\_table

INNER JOIN driver\_dimension ON trip\_fact\_table.Driver\_ID = driver\_dimension.Driver\_ID

WHERE MONTH(trip\_fact\_table.Trip\_Date) = '3' AND YEAR(trip\_fact\_table.Trip\_Date) = '2023'

GROUP BY driver\_dimension.Driver\_Name;

b. Calculate the total revenue earned by the company in a particular week:

SELECT SUM(payment\_fact\_table.Payment\_Amount) AS Total\_Revenue

FROM payment\_fact\_table

INNER JOIN trip\_fact\_table ON payment\_fact\_table.Trip\_ID = trip\_fact\_table.Trip\_ID

WHERE WEEK(trip\_fact\_table.Trip\_Date) = '13' AND YEAR(trip\_fact\_table.Trip\_Date) = '2023';

**Q4. Design a Data Warehouse for Restaurant table booking app like Dineout (Asked in**

**McKinsey for Consultant Data Engineer role)**

**ANSWER:**

Designing a Data Warehouse for a restaurant table booking app like Dineout involves collecting data about restaurants, customers, reservations, orders, and various metrics related to these entities. Here is a possible design for the Data Warehouse:

**Fact Tables:**

a. Reservation Fact Table:

* Reservation\_ID (PK)
* Restaurant\_ID (FK)
* Customer\_ID (FK)
* Reservation\_Date
* Reservation\_Time
* Reservation\_Status
* Table\_Number

b. Order Fact Table:

* Order\_ID (PK)
* Reservation\_ID (FK)
* Restaurant\_ID (FK)
* Customer\_ID (FK)
* Order\_Date
* Order\_Time
* Order\_Status
* Order\_Amount

**Dimension Tables:**

a. Restaurant Dimension Table:

* Restaurant\_ID (PK)
* Restaurant\_Name
* Restaurant\_Address
* Restaurant\_Cuisine
* Restaurant\_Rating

b. Customer Dimension Table:

* Customer\_ID (PK)
* Customer\_Name
* Customer\_Email
* Customer\_Phone
* Customer\_Address

c. Table Dimension Table:

* Table\_Number (PK)
* Restaurant\_ID (FK)

d. Reservation Status Dimension Table:

* Reservation\_Status\_ID (PK)
* Reservation\_Status\_Name

e. Order Status Dimension Table:

* Order\_Status\_ID (PK)
* Order\_Status\_Name

**Primary & Foreign Keys:**

* Reservation\_ID is the Primary Key of Reservation Fact Table and Foreign Key in Order Fact Table.
* Restaurant\_ID, Customer\_ID, and Table\_Number are Foreign Keys in Reservation Fact Table and Primary Keys in the respective Dimension Tables.
* Restaurant\_ID and Customer\_ID are Foreign Keys in Order Fact Table and Primary Keys in the respective Dimension Tables.

**Star Schema Design:**

* The Fact Tables (Reservation and Order) are connected to Dimension Tables (Restaurant, Customer, Table, Reservation Status, and Order Status) through the Primary and Foreign keys.
* The design follows the Star Schema Design, with the Fact Tables forming the center of the schema and the Dimension Tables forming the surrounding points.

**Sample SQL Queries:**

a. Calculate the number of reservations made for each restaurant in a particular month:

SELECT restaurant\_dimension.Restaurant\_Name, COUNT(reservation\_fact\_table.Reservation\_ID) AS Total\_Reservations

FROM reservation\_fact\_table

INNER JOIN restaurant\_dimension ON reservation\_fact\_table.Restaurant\_ID = restaurant\_dimension.Restaurant\_ID

WHERE MONTH(reservation\_fact\_table.Reservation\_Date) = '3' AND YEAR(reservation\_fact\_table.Reservation\_Date) = '2023'

GROUP BY restaurant\_dimension.Restaurant\_Name;

b. Calculate the total revenue earned by a restaurant in a particular week:

SELECT SUM(order\_fact\_table.Order\_Amount) AS Total\_Revenue

FROM order\_fact\_table

INNER JOIN reservation\_fact\_table ON order\_fact\_table.Reservation\_ID = reservation\_fact\_table.Reservation\_ID

WHERE reservation\_fact\_table.Restaurant\_ID = '123' AND WEEK(reservation\_fact\_table.Reservation\_Date) = '13' AND YEAR(reservation\_fact\_table.Reservation\_Date) = '2023';

**Q5. Design a Data Warehouse for Covid Vaccination Application (Asked in Livsapce for**

**Data Engineer role)**

**ANSWER:**

Designing a Data Warehouse for a COVID-19 vaccination application involves collecting data about vaccination centres, vaccination appointments, vaccination doses, and various metrics related to these entities. Here is a possible design for the Data Warehouse:

**Fact Tables:**

a. Vaccination Fact Table:

* Vaccination\_ID (PK)
* Vaccination\_Center\_ID (FK)
* Patient\_ID (FK)
* Vaccination\_Date
* Dose\_Number
* Vaccine\_Type
* Vaccination\_Status

b. Appointment Fact Table

* Appointment\_ID (PK)
* Vaccination\_Center\_ID (FK)
* Patient\_ID (FK)
* Appointment\_Date
* Appointment\_Time
* Appointment\_Status

**Dimension Tables:**

a. Vaccination Center Dimension Table:

* Vaccination\_Center\_ID (PK)
* Center\_Name
* Center\_Address
* Center\_City
* Center\_State

b. Patient Dimension Table:

* Patient\_ID (PK)
* Patient\_Name
* Patient\_Email
* Patient\_Phone
* Patient\_Address

c. Vaccine Dimension Table:

* Vaccine\_Type (PK)
* Manufacturer
* Dose\_Number
* Dose\_Interval

d. Vaccination Status Dimension Table:

* Vaccination\_Status\_ID (PK)
* Vaccination\_Status\_Name

e. Appointment Status Dimension Table:

* Appointment\_Status\_ID (PK)
* Appointment\_Status\_Name

**Primary & Foreign Keys:**

* Vaccination\_ID is the Primary Key of Vaccination Fact Table and Foreign Key in Appointment Fact Table.
* Vaccination\_Center\_ID and Patient\_ID are Foreign Keys in both the Vaccination and Appointment Fact Tables and Primary Keys in the respective Dimension Tables.
* Vaccine\_Type is the Primary Key in the Vaccine Dimension Table and Foreign Key in the Vaccination Fact Table.
* Vaccination\_Status\_ID and Appointment\_Status\_ID are Primary Keys in the respective Dimension Tables and Foreign Keys in both the Vaccination and Appointment Fact Tables.

**Star Schema Design:**

* The Fact Tables (Vaccination and Appointment) are connected to Dimension Tables (Vaccination Center, Patient, Vaccine, Vaccination Status, and Appointment Status) through the Primary and Foreign keys.
* The design follows the Star Schema Design, with the Fact Tables forming the center of the schema and the Dimension Tables forming the surrounding points.

**Sample SQL Queries:**

a. Calculate the number of vaccinations administered by each center in a particular week:

SELECT vaccination\_center\_dimension.Center\_Name, COUNT(vaccination\_fact\_table.Vaccination\_ID) AS Total\_Vaccinations

FROM vaccination\_fact\_table

INNER JOIN vaccination\_center\_dimension ON vaccination\_fact\_table.Vaccination\_Center\_ID = vaccination\_center\_dimension.Vaccination\_Center\_ID

WHERE WEEK(vaccination\_fact\_table.Vaccination\_Date) = '13' AND YEAR(vaccination\_fact\_table.Vaccination\_Date) = '2023'

GROUP BY vaccination\_center\_dimension.Center\_Name;

b. Calculate the percentage of patients who missed their appointments for a particular center:

SELECT (COUNT(appointment\_fact\_table.Appointment\_ID) - COUNT(vaccination\_fact\_table.Vaccination\_ID)) / COUNT(appointment\_fact\_table.Appointment\_ID) \* 100 AS Missed\_Appointments\_Percentage

FROM appointment\_fact\_table

LEFT JOIN vaccination\_fact\_table ON appointment\_fact\_table.Appointment\_ID = vaccination\_fact\_table.Vaccination\_ID

WHERE appointment\_fact\_table.Vaccination\_Center\_ID = '123' AND MONTH(appointment\_fact\_table.Appointment\_Date) = '3' AND YEAR(appointment\_fact\_table.Appointment\_Date) = '2023';