STAR SCHEMA RELATIONSHIP TEACHING NOTES

Complete Guide for All 5 Datasets

With Step-by-Step Power BI Instructions

These teaching notes provide complete instructions for creating Star Schema relationships in Power BI using 5 real-world Indian datasets.

Perfect for classroom or self-study learning!

INTRODUCTION TO STAR SCHEMA (RECAP)

What is Star Schema?

A database design pattern with:

- 1 Central FACT TABLE (large, contains transactions/measurements)
- Multiple DIMENSION TABLES (small, contain attributes/descriptions)
- ONE-TO-MANY RELATIONSHIPS between fact and each dimension
- Fact table in center, dimensions radiating outward (looks like a star)

Why Learn This?

✓ Professional BI standard (used by all companies)
✓ Best performance for reporting
✓ Easy to understand and maintain
✓ Essential for Power BI modeling
✓ Foundation for advanced analytics

Key Components:

FACT TABLE:

- Large table (100s to millions of rows)
- Contains measurements/amounts (sales, costs, marks, fares)
- Contains Foreign Keys (StoreID, ProductID, CustomerID, DateID)
- One row per transaction/event
- Updated frequently with new data
- Example: SalesData, PatientTreatment, StudentGrades

DIMENSION TABLE:

- Small table (10s to 1000s of rows)
- Contains descriptions/attributes (names, categories, cities)
- Contains Primary Key (unique identifier)
- Rarely changes (static or slowly changing)
- Used for filtering and grouping
- Example: StoreDim, ProductDim, CustomerDim, DateDim

RELATIONSHIP:

- Direction: Dimension (One) ⟷ Fact (Many)
- Type: One-to-Many
- Meaning: One store has many sales, one product has many sales, etc.

DATASET 1: RETAIL SALES STAR SCHEMA

FILE: Star_Schema_1_Retail_Sales.xlsx

BUSINESS SCENARIO:

Indian retail chain with 5 stores in major cities
Selling 8 different products (electronics, clothing, appliances)
Tracking daily sales with customer information
Want to analyze sales by store, product, customer, and time period

FACT TABLE: SalesData

Table Details:

- 100 rows (each row = 1 sale transaction)
- Primary Key: SalesID (1-100, unique identifier)
- Foreign Keys: StoreID, ProductID, CustomerID, DateID
- Measurements: Quantity (number of items), Amount (price in rupees)

Columns:

```
  SalesID       - Unique sale identifier (1, 2, 3... 100)
  StoreID       - Links to StoreDim (101, 102, 103, 104, 105)
  ProductID     - Links to ProductDim (1001-1008)
  CustomerID    - Links to CustomerDim (5001-5008)
  DateID        - Links to DateDim (1-365)
  Quantity      - Number of items sold (1-5)
  Amount        - Total rupees for this sale (₹500 to ₹500,000)
```

Example Data:

SalesID	StoreID	ProductID	CustomerID	DateID	Qty	Amount
1	101	1001	5001	1	2	15000
2	102	1002	5002	1	1	45000
3	101	1003	5001	2	5	25000

DIMENSION TABLE 1: StoreDim

Table Details:

- 5 rows (one per store)
- Primary Key: StoreID (101, 102, 103, 104, 105)
- Attributes: StoreName, City, Manager

Real Data:

StoreID	StoreName	City	Manager
101	Mumbai Store	Mumbai	Rajesh Kumar
102	Delhi Store	Delhi	Priya Singh
103	Bangalore Store	Bangalore	Amit Patel
104	Chennai Store	Chennai	Neha Gupta
105	Hyderabad Store	Hyderabad	Vikram Sharma

Purpose:

- Identify which store the sale occurred at
- Filter sales by city
- Analyze performance by store manager
- Group sales by store location

DIMENSION TABLE 2: ProductDim

Table Details:

- 8 rows (one per product)
- Primary Key: ProductID (1001-1008)
- Attributes: ProductName, Category, Price

Real Data:

ProductID	ProductName	Category	Price
1001	Samsung TV	Electronics	45000
1002	Laptop	Electronics	75000
1003	T-Shirt	Clothing	500
1004	Jeans	Clothing	1500
1005	Microwave	Home Appliances	8000
1006	Fan	Home Appliances	3000
1007	Shoes	Clothing	2000
1008	Bedsheet	Home Textiles	1000

Purpose:

- Identify which product was sold
- Group sales by category
- Analyze sales by product price range
- Filter by specific product names

DIMENSION TABLE 3: CustomerDim

Table Details:

- 8 rows (one per customer)
- Primary Key: CustomerID (5001-5008)
- Attributes: Name, City, Segment (Premium/Regular/Budget)

Real Data:

CustomerID	Name	City	Segment
5001	Rajesh K	Mumbai	Premium
5002	Priya S	Delhi	Regular
5003	Amit P	Bangalore	Regular
5004	Neha G	Chennai	Premium
5005	Vikram S	Hyderabad	Budget
5006	Sakshi K	Mumbai	Regular
5007	Rohan M	Pune	Budget
5008	Anjali R	Kolkata	Premium

Purpose:

- Identify customer who made the purchase
- Filter sales by customer segment
- Analyze purchasing patterns by segment
- Calculate customer lifetime value

DIMENSION TABLE 4: DateDim

Table Details:

- 365 rows (one per day in 2025)
- Primary Key: DateID (1-365)
- Attributes: Date, Month, MonthNum, Quarter, Year

Sample Data:

DateID	Date	Month	MonthNum	Quarter	Year
1	2025-01-01	January	1	Q1	2025
2	2025-01-02	January	1	Q1	2025
...					
365	2025-12-31	December	12	Q4	2025

Purpose:

- Convert numeric DateID to actual dates
- Filter by month, quarter, or year
- Calculate time-based metrics (YTD, QTD, MTD)
- Analyze seasonal trends

STEP-BY-STEP POWER BI INSTRUCTIONS:

STEP 1: Open Power BI Desktop

1. Launch Power BI Desktop
2. Click "Get Data" (Home tab)
3. Select "Excel"
4. Navigate to Star_Schema_1_Retail_Sales.xlsx
5. Click Open

STEP 2: Select Tables to Import

1. In Navigator window, you see all 5 sheets

```
  2. Check these boxes:
     ✓ SalesData (Fact table)
     ✓ StoreDim
     ✓ ProductDim
     ✓ CustomerDim
     ✓ DateDim
  3. Click "Load"
  4. Wait for data to import

STEP 3: Go to Model View
  1. In Power BI ribbon, click "Model" (left view options)
  2. You now see all 5 tables in canvas
  3. Tables appear as boxes with column names

STEP 4: Create Relationship 1 - Store
  1. In Model view, find SalesData table
  2. Find StoreID column in SalesData
  3. Drag SalesData.StoreID to StoreDim.StoreID
  4. Dialog appears asking for relationship details
  5. Verify:
     - From table: SalesData
     - From column: StoreID
     - To table: StoreDim
     - To column: StoreID
     - Direction: Single (StoreDim One → SalesData Many)
  6. Click OK

Expected Result:
  A line appears connecting SalesData to StoreDim
  StoreDim has "1" marker (one side)
  SalesData has "*" marker (many side)

STEP 5: Create Relationship 2 - Product
  1. Drag SalesData.ProductID to ProductDim.ProductID
  2. Verify direction: ProductDim (One) → SalesData (Many)
  3. Click OK

STEP 6: Create Relationship 3 - Customer
  1. Drag SalesData.CustomerID to CustomerDim.CustomerID
  2. Verify direction: CustomerDim (One) → SalesData (Many)
  3. Click OK

STEP 7: Create Relationship 4 - Date
  1. Drag SalesData.DateID to DateDim.DateID
  2. Verify direction: DateDim (One) → SalesData (Many)
  3. Click OK

STEP 8: Mark Date Table
  1. Right-click DateDim table
  2. Select "Mark as date table"
  3. A dialog appears
  4. Confirm DateDim[Date] is selected as date column
  5. Click OK
  6. DateDim icon changes (indicates it's a date table)

STEP 9: Verify Star Schema
  1. Look at Model view canvas
  2. You should see STAR SHAPE:
     - SalesData in center
     - 4 dimension tables radiating outward
     - All connected with lines
  3. All relationships show correct direction (One to Many)

STEP 10: Create Sample Visuals
```

1. Click "Report" tab
2. Click "Insert" → "Card"
3. Drag SalesData[Amount] to card visual
4. Shows total: ₹33,900,000 (sum of all sales)

5. Insert Table visual
6. Add fields:
   - StoreDim[StoreName] (rows)
   - SalesData[Amount] (values/sum)
7. Shows sales by store

8. Insert another Table
9. Add fields:
   - ProductDim[Category] (rows)
   - SalesData[Amount] (values/sum)
10. Shows sales by product category

STEP 11: Add Slicers

1. Insert Slicer visual
2. Add StoreDim[City] to slicer
3. Shows all 5 cities: Mumbai, Delhi, Bangalore, Chennai, Hyderabad
4. When you click a city, all visuals filter to that city

5. Insert another Slicer
6. Add DateDim[Month] to slicer
7. Shows all 12 months
8. When you click a month, all visuals filter to that month

KEY TAKEAWAYS FOR DATASET 1:

✓ Fact table (SalesData) contains transactions
✓ 4 dimension tables contain attributes
✓ All relationships are One-to-Many
✓ StoreDim has 5 unique stores
✓ ProductDim has 8 unique products
✓ CustomerDim has 8 unique customers
✓ DateDim has 365 unique dates
✓ Star schema allows easy filtering and analysis
✓ All tables linked by foreign keys
✓ Data is clean and ready for analysis

COMMON QUESTIONS:

Q: Why do we have separate tables instead of one big table?
A: Separate tables allow:
- Better performance (smaller tables, faster queries)
- Less data duplication (store name only once, not 20 times)
- Easier updates (change store manager in one place)
- Flexible reporting (group by any dimension)

Q: What does "One-to-Many" mean?
A: One store can have MANY sales
One product can be in MANY sales
One customer can make MANY purchases
One date can have MANY transactions

Q: How do I know relationships are correct?
A: Check these:
- Direction shows 1 on dimension side, * on fact side
- Primary key in dimension matches foreign key in fact
- No errors appear in Power BI
- Filtering works as expected in reports

Q: What if I accidentally delete a relationship?

A: Go to Model view, create relationship again by dragging

DATASET 2: HOSPITAL PATIENT CARE STAR SCHEMA

FILE: Star_Schema_2_Hospital_Care.xlsx

BUSINESS SCENARIO:
Indian hospital tracking patient treatments
Multiple doctors, departments, and patients
Want to analyze treatment costs, doctor performance, department workload
Track patient care across time

FACT TABLE: PatientTreatment

Table Details:

- 100 rows (each row = 1 treatment record)
- Primary Key: TreatmentID (1-100)
- Foreign Keys: PatientID, DoctorID, DeptID, DateID
- Measurements: Cost (treatment cost in rupees)

Columns:
TreatmentID - Unique treatment ID (1-100)
PatientID - Links to PatientDim (P001-P100)
DoctorID - Links to DoctorDim (D101-D110)
DeptID - Links to DepartmentDim (10, 20, 30, 40, 50)
DateID - Links to DateDim (1-365)
Cost - Cost of treatment (₹2000-₹25000)

Example Data:

TreatmentID	PatientID	DoctorID	DeptID	DateID	Cost
1	P001	D101	10	1	5000
2	P002	D102	20	1	8000
3	P001	D101	10	2	3000

DIMENSION TABLE 1: PatientDim

Table Details:

- 20 rows (one per patient)
- Primary Key: PatientID (P001-P020)
- Attributes: PatientName, Age, City

Purpose:

- Identify patient receiving treatment
- Filter by city
- Analyze by age group
- Track patient history

DIMENSION TABLE 2: DoctorDim

Table Details:

- 10 rows (one per doctor)
- Primary Key: DoctorID (D101-D110)
- Attributes: DoctorName, Specialty

Sample Data:

DoctorID	DoctorName	Specialty
D101	Dr. Sharma	Cardiology
D102	Dr. Patel	Neurology
D103	Dr. Gupta	Pediatrics
...		

Purpose:

- Identify treating doctor
- Analyze doctor workload
- Compare doctor performance
- Group by specialty

DIMENSION TABLE 3: DepartmentDim

Table Details:

- 5 rows (one per department)
- Primary Key: DeptID (10, 20, 30, 40, 50)
- Attributes: DeptName, Ward

Real Data:

DeptID	DeptName	Ward
10	Cardiology	Ward-A
20	Neurology	Ward-B
30	Pediatrics	Ward-C
40	Orthopedics	Ward-D
50	Dermatology	Ward-E

Purpose:

- Identify treatment department
- Analyze departmental costs
- Compare ward performance
- Track department capacity

DIMENSION TABLE 4: DateDim

Table Details:

- 365 rows (one per day in 2025)
- Primary Key: DateID (1-365)
- Attributes: Date, Month, Quarter, Year

Purpose:

- Convert DateID to actual dates
- Filter by time period
- Analyze seasonal treatment patterns
- Calculate monthly/quarterly costs

STEP-BY-STEP POWER BI INSTRUCTIONS:

STEP 1: Import Data

1. Get Data → Excel
2. Open Star_Schema_2_Hospital_Care.xlsx
3. Select all 5 sheets:
   ✓ PatientTreatment
   ✓ PatientDim
   ✓ DoctorDim
   ✓ DepartmentDim
   ✓ DateDim
4. Click Load

STEP 2: Go to Model View

1. Click Model tab
2. See all 5 tables on canvas

STEP 3: Create Relationships

Relationship 1 - Patient:

Drag PatientTreatment.PatientID to PatientDim.PatientID

Relationship 2 - Doctor:
Drag PatientTreatment.DoctorID to DoctorDim.DoctorID

Relationship 3 - Department:
Drag PatientTreatment.DeptID to DepartmentDim.DeptID

Relationship 4 - Date:
Drag PatientTreatment.DateID to DateDim.DateID

STEP 4: Mark Date Table

1. Right-click DateDim
2. Mark as date table
3. Confirm DateDim[Date] is selected

STEP 5: Verify Star Schema

1. Check Model view shows star shape
2. PatientTreatment in center
3. 4 dimensions radiating outward

STEP 6: Create Sample Report

1. Go to Report tab
2. Create Card: Sum of PatientTreatment[Cost]
   Result: Total healthcare cost

3. Create Table:
   Rows: DoctorDim[DoctorName]
   Values: PatientTreatment[Cost] (sum)
   Result: Cost per doctor

4. Create Table:
   Rows: DepartmentDim[DeptName]
   Values: PatientTreatment[Cost] (sum)
   Result: Cost by department

5. Create Slicer:
   DepartmentDim[DeptName]
   Result: Filter all visuals by department

SAMPLE INSIGHTS YOU CAN CREATE:

✓ Total treatment cost: ₹X,XXX,XXX
✓ Average cost per treatment: ₹X,XXX
✓ Cardiology costs: ₹X,XXX,XXX (most expensive)
✓ Dr. Sharma's patients treated: 10+
✓ Ward-A capacity: X patients
✓ Monthly cost trends
✓ Doctor specialization performance
✓ Patient city distribution

KEY TAKEAWAYS FOR DATASET 2:

✓ Healthcare data uses same star schema pattern
✓ Facts are treatments (costs)
✓ Dimensions are patients, doctors, departments, dates
✓ One doctor treats many patients
✓ One department handles many treatments
✓ One patient receives multiple treatments
✓ Star schema works across different industries

DATASET 3: EDUCATIONAL INSTITUTION STAR SCHEMA

FILE: Star_Schema_3_Education.xlsx

BUSINESS SCENARIO:
School/college tracking student performance
Multiple students, courses, teachers, semesters
Want to analyze grades, teacher effectiveness, course difficulty
Track academic progress over time

FACT TABLE: StudentGrades

Table Details:

- 100 rows (each row = 1 grade record)
- Primary Key: GradeID (1-100)
- Foreign Keys: StudentID, CourseID, TeacherID, SemesterID
- Measurements: Marks (0-100, student score)

Columns:
GradeID - Unique grade ID (1-100)
StudentID - Links to StudentDim (S001-S030)
CourseID - Links to CourseDim (C101-C106)
TeacherID - Links to TeacherDim (T101-T110)
SemesterID - Links to SemesterDim (1, 2)
Marks - Student score (0-100)

Example Data:

GradeID	StudentID	CourseID	TeacherID	SemesterID	Marks
1	S001	C101	T101	1	85
2	S002	C101	T101	1	92
3	S001	C102	T102	1	78

DIMENSION TABLE 1: StudentDim

Table Details:

- 30 rows (one per student)
- Primary Key: StudentID (S001-S030)
- Attributes: StudentName, Class, City

Purpose:

- Identify student
- Group by class (10A, 10B)
- Filter by city
- Track individual performance

DIMENSION TABLE 2: CourseDim

Table Details:

- 6 rows (one per course)
- Primary Key: CourseID (C101-C106)
- Attributes: CourseName, Subject, MaxMarks

Real Data:

CourseID	CourseName	Subject	MaxMarks
C101	English	Literature	100
C102	Mathematics	Calculus	100
C103	Science	Physics	100
C104	Hindi	Grammar	100
C105	Social Studies	History	100
C106	Computer Science	Programming	100

Purpose:

- Identify course
- Analyze by subject

- Compare course difficulty
- Track subject-wise performance

DIMENSION TABLE 3: TeacherDim

Table Details:

- 10 rows (one per teacher)
- Primary Key: TeacherID (T101-T110)
- Attributes: TeacherName, Subject, Experience

Purpose:

- Identify teaching teacher
- Analyze by subject taught
- Compare teacher effectiveness
- Consider experience level

DIMENSION TABLE 4: SemesterDim

Table Details:

- 2 rows (Semester 1, Semester 2)
- Primary Key: SemesterID (1, 2)
- Attributes: SemesterName, StartDate, EndDate

Real Data:

SemesterID	SemesterName	StartDate	EndDate
1	Semester 1	2025-01-15	2025-04-15
2	Semester 2	2025-05-01	2025-07-31

Purpose:

- Identify academic period
- Compare semesters
- Track progress over time

STEP-BY-STEP POWER BI INSTRUCTIONS:

STEP 1: Import Data

1. Get Data → Excel
2. Open Star_Schema_3_Education.xlsx
3. Select all sheets:
   ✓ StudentGrades
   ✓ StudentDim
   ✓ CourseDim
   ✓ TeacherDim
   ✓ SemesterDim
4. Click Load

STEP 2: Go to Model View

1. Click Model tab

STEP 3: Create Relationships

Relationship 1 - Student:
Drag StudentGrades.StudentID to StudentDim.StudentID

Relationship 2 - Course:
Drag StudentGrades.CourseID to CourseDim.CourseID

Relationship 3 - Teacher:
Drag StudentGrades.TeacherID to TeacherDim.TeacherID

Relationship 4 - Semester:
Drag StudentGrades.SemesterID to SemesterDim.SemesterID

STEP 4: Verify Star Schema

1. StudentGrades in center
2. 4 dimensions radiating outward

STEP 5: Create Sample Report

1. Card: Average of StudentGrades[Marks]
   Result: Class average (e.g., 78.5)

2. Table: CourseDim[CourseName], StudentGrades[Marks] (average)
   Result: Average score by course

3. Table: TeacherDim[TeacherName], StudentGrades[Marks] (average)
   Result: Teacher effectiveness (student average)

4. Table: StudentDim[StudentName], StudentGrades[Marks]
   Result: Student report card

5. Slicer: SemesterDim[SemesterName]
   Result: Compare Semester 1 vs Semester 2

SAMPLE INSIGHTS YOU CAN CREATE:

✓ Class average: 78.5 marks
✓ Highest scorer: 100 marks
✓ Subject average: Math (75), English (82), Science (80)
✓ Teacher effectiveness: Which teacher gets highest avg grades
✓ Course difficulty: Which course has lowest average
✓ Student performance: Who's struggling, who's excelling
✓ Semester comparison: Improvement from Sem1 to Sem2
✓ Class distribution: 10A vs 10B average

KEY TAKEAWAYS FOR DATASET 3:

✓ Education sector uses same star schema
✓ Facts are grades (marks)
✓ Dimensions are students, courses, teachers, semesters
✓ One teacher teaches multiple courses/students
✓ One course has multiple students
✓ One student takes multiple courses
✓ Easy to compare performance across dimensions

DATASET 4: AIRLINE BOOKING STAR SCHEMA

FILE: Star_Schema_4_Airline_Booking.xlsx

BUSINESS SCENARIO:
Indian airline tracking flight bookings
Multiple flights, passengers, airports, airlines
Want to analyze revenue, route performance, seat utilization
Track booking trends over time

FACT TABLE: FlightBooking

Table Details:

- 100 rows (each row = 1 booking)
- Primary Key: BookingID (1-100)
- Foreign Keys: PassengerID, FlightID, AirportID, DateID
- Measurements: Fare (ticket price in rupees)

Columns:

```
  BookingID    - Unique booking ID (1-100)
```

PassengerID - Links to PassengerDim (P0001-P0030)
FlightID - Links to FlightDim (F101-F110)
AirportID - Links to AirportDim (A001-A005)
DateID - Links to DateDim (1-365)
Fare - Ticket price (₹5000-₹25000)

Example Data:

BookingID	PassengerID	FlightID	AirportID	DateID	Fare
1	P0001	F101	A001	1	8000
2	P0002	F101	A001	1	7500
3	P0003	F102	A002	2	5000

DIMENSION TABLE 1: PassengerDim

Table Details:

- 30 rows (one per passenger)
- Primary Key: PassengerID (P0001-P0030)
- Attributes: PassengerName, City, Class (Business/Economy)

Purpose:

- Identify passenger
- Filter by city
- Analyze by cabin class
- Track frequent flyers

DIMENSION TABLE 2: FlightDim

Table Details:

- 10 rows (one per flight)
- Primary Key: FlightID (F101-F110)
- Attributes: FlightNumber, AirlineName, Capacity

Real Data:

FlightID	FlightNumber	AirlineName	Capacity
F101	AI-101	Air India	180
F102	AI-202	Air India	200
F103	SG-303	Spicejet	150
F104	I6-404	Indigo	160
...			

Purpose:

- Identify flight
- Analyze by airline
- Compare capacity utilization
- Track route performance

DIMENSION TABLE 3: AirportDim

Table Details:

- 5 rows (one per airport)
- Primary Key: AirportID (A001-A005)
- Attributes: AirportName, City, IATA

Real Data:

AirportID	AirportName	City	IATA
A001	Indira Gandhi	Delhi	DEL
A002	Bombay Inter.	Mumbai	BOM
A003	Kempegowda	Bangalore	BLR
A004	Rajiv Gandhi	Hyderabad	HYD
A005	Cochin Inter.	Kochi	COK

Purpose:

- Identify airport (destination)
- Analyze by city
- Compare airport revenue
- Track route popularity

DIMENSION TABLE 4: DateDim

Table Details:

- 365 rows (one per day)
- Primary Key: DateID (1-365)
- Attributes: Date, Month, Quarter, Year

Purpose:

- Track booking date
- Identify seasonal trends
- Compare monthly revenue
- Analyze year-round patterns

STEP-BY-STEP POWER BI INSTRUCTIONS:

STEP 1: Import Data

1. Get Data → Excel
2. Open Star_Schema_4_Airline_Booking.xlsx
3. Select all sheets:
   ✓ FlightBooking
   ✓ PassengerDim
   ✓ FlightDim
   ✓ AirportDim
   ✓ DateDim
4. Click Load

STEP 2: Go to Model View

1. Click Model tab

STEP 3: Create Relationships

Relationship 1 - Passenger:
Drag FlightBooking.PassengerID to PassengerDim.PassengerID

Relationship 2 - Flight:
Drag FlightBooking.FlightID to FlightDim.FlightID

Relationship 3 - Airport:
Drag FlightBooking.AirportID to AirportDim.AirportID

Relationship 4 - Date:
Drag FlightBooking.DateID to DateDim.DateID

STEP 4: Mark Date Table

1. Right-click DateDim
2. Mark as date table

STEP 5: Verify Star Schema

1. FlightBooking in center
2. 4 dimensions radiating outward

STEP 6: Create Sample Report

1. Card: Sum of FlightBooking[Fare]
   Result: Total revenue

2. Table: FlightDim[FlightNumber], FlightBooking[Fare] (sum)
   Result: Revenue by flight

3. Table: AirportDim[City], FlightBooking[Fare] (sum)
   Result: Revenue by airport/city

4. Table: PassengerDim[Class], FlightBooking[Fare] (sum)
   Result: Business vs Economy revenue

5. Slicer: DateDim[Month]
   Result: Filter by month

SAMPLE INSIGHTS YOU CAN CREATE:

✓ Total revenue: ₹X,XXX,XXX
✓ Average fare: ₹X,XXX per ticket
✓ Top airport: Mumbai with ₹X,XXX,XXX revenue
✓ Business vs Economy: Revenue split
✓ Busiest airline: Air India
✓ Best-selling flight: AI-101
✓ Monthly trends: Peak season bookings
✓ Occupancy rate: Bookings vs Capacity

KEY TAKEAWAYS FOR DATASET 4:

✓ Airline industry uses star schema
✓ Facts are bookings (revenue)
✓ Dimensions are passengers, flights, airports, dates
✓ One flight has multiple passengers
✓ One airport handles multiple flights
✓ One airline operates multiple flights
✓ Easy to analyze revenue across different dimensions

DATASET 5: RESTAURANT CHAIN STAR SCHEMA

FILE: Star_Schema_5_Restaurant_Orders.xlsx

BUSINESS SCENARIO:
Indian restaurant chain with 5 outlets in major cities
Serving 15 different menu items
Tracking daily orders from customers
Want to analyze sales, popular items, location performance

FACT TABLE: OrderData

Table Details:
- 100 rows (each row = 1 order line)
- Primary Key: OrderID (1-100)
- Foreign Keys: MenuItemID, CustomerID, RestaurantID, DateID
- Measurements: Quantity, Amount (in rupees)

Columns:
OrderID - Unique order ID (1-100)
MenuItemID - Links to MenuItemDim (M1001-M1015)
CustomerID - Links to CustomerDim (C1001-C1025)
RestaurantID - Links to RestaurantDim (R1001-R1005)
DateID - Links to DateDim (1-365)
Quantity - Items ordered (1-5)
Amount - Total rupees (₹50-₹6000)

Example Data:

OrderID	MenuItemID	CustomerID	RestaurantID	DateID	Qty	Amount
1	M1001	C1001	R1001	1	2	800
2	M1002	C1002	R1001	1	1	1200

3	M1003	C1003	R1002	2	3	450

DIMENSION TABLE 1: MenuItemDim

Table Details:

- 15 rows (one per menu item)
- Primary Key: MenuItemID (M1001-M1015)
- Attributes: ItemName, Category, Price

Real Data:

MenuItemID	ItemName	Category	Price
M1001	Butter Chicken	Main Course	400
M1002	Paneer Tikka	Appetizer	300
M1003	Samosa	Appetizer	80
M1004	Biryani	Main Course	250
M1005	Dosa	Breakfast	120
...			

Purpose:

- Identify menu item ordered
- Group by category
- Analyze by price range
- Track popular items

DIMENSION TABLE 2: CustomerDim

Table Details:

- 25 rows (one per customer)
- Primary Key: CustomerID (C1001-C1025)
- Attributes: CustomerName, City, MemberType (Premium/Regular/VIP)

Purpose:

- Identify customer
- Filter by city
- Analyze by membership level
- Track customer preferences

DIMENSION TABLE 3: RestaurantDim

Table Details:

- 5 rows (one per restaurant)
- Primary Key: RestaurantID (R1001-R1005)
- Attributes: RestaurantName, City, Manager

Real Data:

RestaurantID	RestaurantName	City	Manager
R1001	Mumbai North	Mumbai	Rajesh Singh
R1002	Delhi Central	Delhi	Priya Sharma
R1003	Bangalore South	Bangalore	Amit Kumar
R1004	Chennai East	Chennai	Neha Patel
R1005	Hyderabad West	Hyderabad	Vikram Reddy

Purpose:

- Identify restaurant outlet
- Analyze by location
- Compare outlet performance
- Track manager performance

DIMENSION TABLE 4: DateDim

Table Details:

- 365 rows (one per day)
- Primary Key: DateID (1-365)
- Attributes: Date, Month, Quarter, Year

Purpose:

- Track order date
- Identify daily/weekly trends
- Compare month-over-month
- Analyze seasonal patterns

STEP-BY-STEP POWER BI INSTRUCTIONS:

STEP 1: Import Data

1. Get Data → Excel
2. Open Star_Schema_5_Restaurant_Orders.xlsx
3. Select all sheets:
   ✓ OrderData
   ✓ MenuItemDim
   ✓ CustomerDim
   ✓ RestaurantDim
   ✓ DateDim
4. Click Load

STEP 2: Go to Model View

1. Click Model tab

STEP 3: Create Relationships

Relationship 1 - MenuItem:
Drag OrderData.MenuItemID to MenuItemDim.MenuItemID

Relationship 2 - Customer:
Drag OrderData.CustomerID to CustomerDim.CustomerID

Relationship 3 - Restaurant:
Drag OrderData.RestaurantID to RestaurantDim.RestaurantID

Relationship 4 - Date:
Drag OrderData.DateID to DateDim.DateID

STEP 4: Mark Date Table

1. Right-click DateDim
2. Mark as date table

STEP 5: Verify Star Schema

1. OrderData in center
2. 4 dimensions radiating outward

STEP 6: Create Sample Report

1. Card: Sum of OrderData[Amount]
   Result: Total revenue

2. Table: MenuItemDim[ItemName], OrderData[Amount] (sum)
   Result: Revenue by menu item

3. Table: RestaurantDim[RestaurantName], OrderData[Amount] (sum)
   Result: Revenue by outlet

4. Table: MenuItemDim[Category], OrderData[Amount] (sum)
   Result: Revenue by category

5. Slicer: RestaurantDim[City]
   Result: Filter by city

SAMPLE INSIGHTS YOU CAN CREATE:

✓ Total revenue: ₹X,XXX,XXX
✓ Best seller: Butter Chicken (₹X,XXX revenue)
✓ Popular category: Main Course
✓ Best outlet: Mumbai North (₹X,XXX revenue)
✓ Customer segment: Premium vs Regular vs VIP
✓ Daily average: ₹X,XXX per day
✓ Peak month: Which month highest sales
✓ Item average: ₹X per item
✓ City comparison: Mumbai vs Delhi vs Bangalore
✓ Order frequency: How many orders per day

KEY TAKEAWAYS FOR DATASET 5:

✓ Food service industry uses star schema
✓ Facts are orders (revenue)
✓ Dimensions are menu items, customers, restaurants, dates
✓ One menu item ordered many times
✓ One restaurant has many customers
✓ One customer eats at multiple restaurants
✓ Same star pattern applies to all industries

==

COMPARING ALL 5 DATASETS

==

Dataset	Domain	Fact Table	# Dimensions	Key Measure
Retail	Commerce	SalesData	4	Amount (₹)
Hospital	Healthcare	PatientTreatment	4	Cost (₹)
Education	Academia	StudentGrades	4	Marks (0-100)
Airline	Travel	FlightBooking	4	Fare (₹)
Restaurant	Food Service	OrderData	4	Amount (₹)

OBSERVATIONS:

✓ All follow same Star Schema pattern
✓ All have 1 Fact Table + 4 Dimension Tables
✓ All have 100 rows in Fact Table
✓ All have Foreign Keys in Fact Table
✓ All have Primary Keys in Dimension Tables
✓ All relationships are One-to-Many
✓ All have a DateDim for time tracking
✓ All include 365 dates (full year 2025)
✓ Star pattern looks identical in Model view
✓ Same Power BI steps apply to all

==

COMMON MISTAKES AND HOW TO AVOID THEM

==

MISTAKE 1: Wrong Relationship Direction
  Problem: Created Many-to-One instead of One-to-Many
  Symptom: Filtering doesn't work correctly
  Solution:

```
    1. Go to Model view
    2. Double-click the relationship line
    3. Check direction: Dimension (One) → Fact (Many)
    4. If wrong, click Edit and fix direction

MISTAKE 2: Forgot to Mark Date Table
  Problem: Time intelligence functions don't work
  Symptom: Can't filter by month/quarter/year properly
  Solution:
    1. Right-click DateDim table
    2. Select "Mark as date table"
    3. Confirm Date column is selected
    4. Click OK

MISTAKE 3: Created Relationship Between Wrong Columns
  Problem: Some records don't match
  Symptom: Blank values appear in filtered reports
  Solution:
    1. Check if foreign key in Fact matches Primary Key in Dimension
    2. Verify data types match (both should be same type)
    3. Check for spelling differences or extra spaces
    4. Delete relationship and recreate

MISTAKE 4: Two-Way Relationship Created by Accident
  Problem: Filtering works from both directions (incorrect)
  Symptom: Unexpected filter behavior
  Solution:
    1. Go to Model view
    2. Double-click relationship
    3. Change from "Both" to "Single" direction
    4. Make sure it's Single direction: Dimension → Fact

MISTAKE 5: Didn't Check Primary Key is Unique
  Problem: Duplicate rows in dimension cause inflated numbers
  Symptom: Total amounts seem too high
  Solution:
    1. Open dimension table data view
    2. Check Primary Key column (should be all unique)
    3. If duplicates found, delete them
    4. Recalculate report

MISTAKE 6: Relationships Look Messy (Not Star-Shaped)
  Problem: Relationships crossing over each other
  Symptom: Looks confusing, hard to understand
  Solution:
    1. This is just visual - relationships are still correct
    2. Go to Model view
    3. Drag tables to rearrange (put Fact in center)
    4. Arrange Dimensions in circle around Fact
   5. Clean layout makes it easier to teach
```