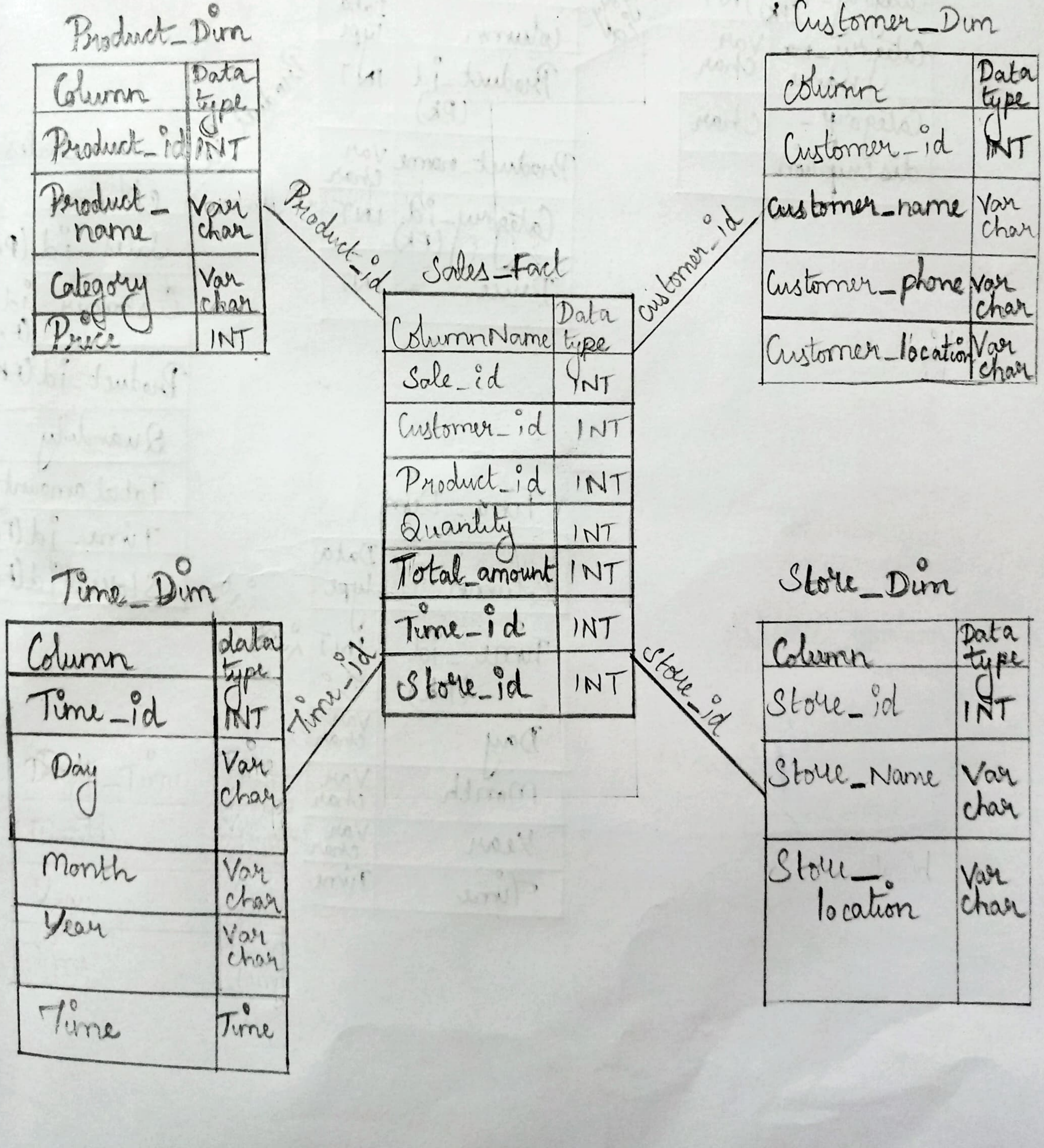
1) Star Schema and Snowflake schema

E-commerce sales:

Star Schema:



Product_Dim

Column	Data type
Product_id	INT
Product_name	Var char
Category	Var char
Price	INT

Customer_Dim

Column	Data type
Customer_id	INT
Customer_name	Var char
Customer_phone	Var char
Customer_location	Var char

Sales_Fact

ColumnName	Data type
Sale_id	INT
Customer_id	INT
Product_id	INT
Quantity	INT
Total_amount	INT
Time_id	INT
Store_id	INT

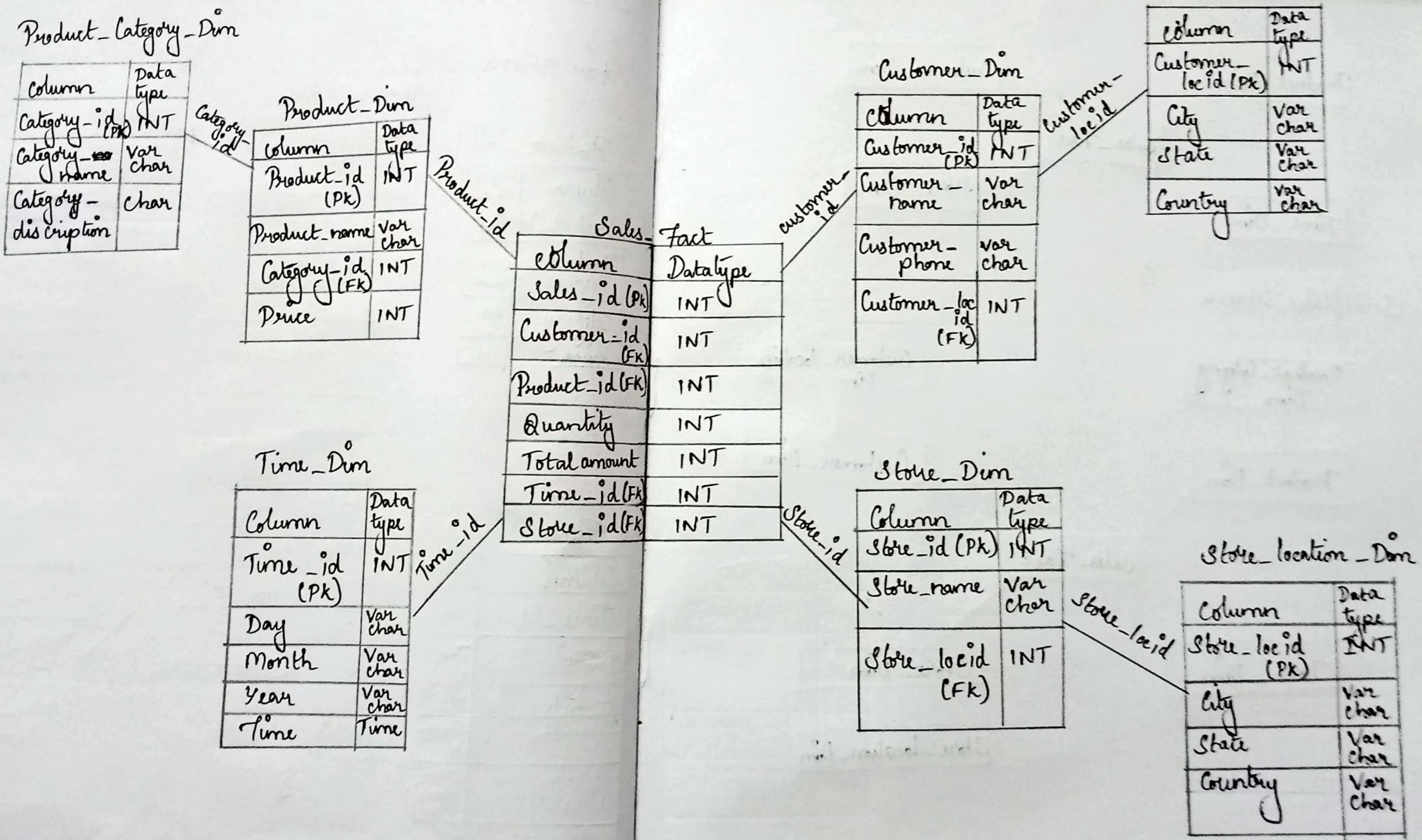
Time_Dim

Column	data type
Time_id	INT
Day	Var char
Month	Var char
Year	Var char
Time	Time

Store_Dim

Column	Data type
Store_id	INT
Store_Name	Var char
Store_location	Var char

Snowflake Schema :



Product_Category_Dim

Column	Data type
Category_id (PK)	INT
Category_name	Var char
Category_discription	Char

Product_Dim

Column	Data type
Product_id (PK)	INT
Product_name	Var char
Category_id (FK)	INT
Price	INT

Sales_Fact

Column	Datatype
Sales_id (Pk)	INT
Customer_id (FK)	INT
Product_id (FK)	INT
Quantity	INT
Total amount	INT
Time_id (FK)	INT
Store_id (FK)	INT

Customer_Dim

Column	Data type
Customer_id (PK)	INT
Customer_name	Var char
Customer_phone	Var char
Customer_loc id (FK)	INT

Customer_location_Dim

Column	Data type
Customer_loc id (PK)	INT
City	Var char
State	Var char
Country	Var char

Time_Dim

Column	Data type
Time_id (PK)	INT
Day	Var char
Month	Var char
Year	Var char
Time	Time

Store_Dim

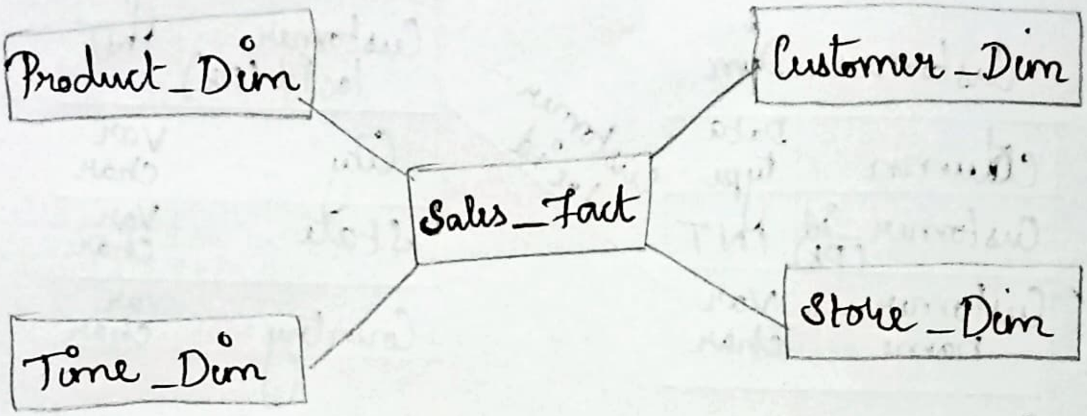
Column	Data type
Store_id (PK)	INT
Store_name	Var char
Store_loc id (FK)	INT

Store_location_Dim

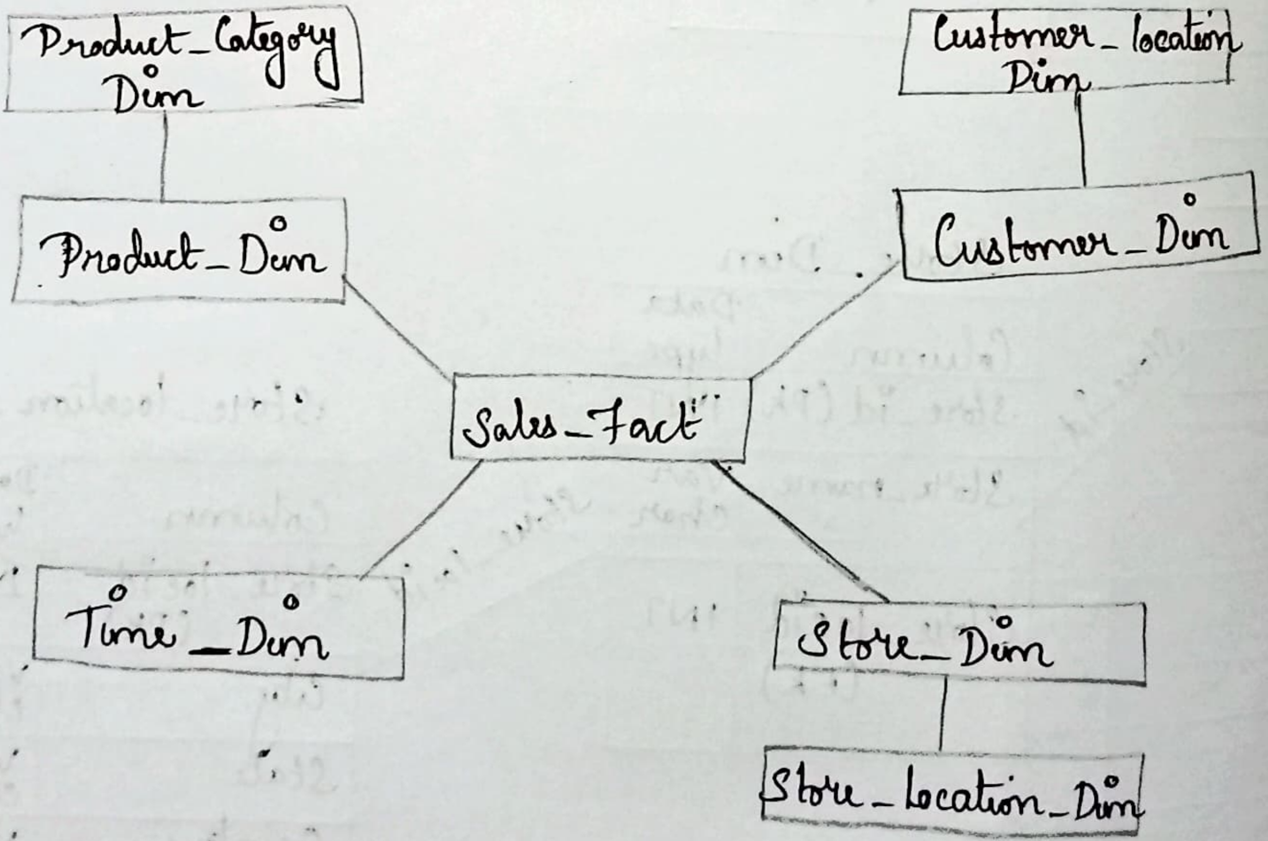
Column	Data type
Store_loc id (PK)	INT
City	Var char
State	Var char
Country	Var char

2) Dimensional structure and Table Relationships

Star schema :



Snowflake Schema



3. Performance and Storage Analysis

Factor	**Star Schema**	**Snowflake Schema**
Query Performance	Faster (fewer joins)	Slower (more joins, more lookups)
Storage	Higher (redundant data in dims)	Lower (normalized data, less redundancy)
Join Complexity	Low (simple star joins)	High (joins across multiple dimension layers)
ETL Time	Faster load, simpler pipelines	More complex load (data splitting)
Data Integrity	Lower (harder to enforce consistency)	Higher (normalized structure enforces rules)

4. Practical Considerations

When to Use Star Schema:

- Business Intelligence and OLAP systems
- Use cases prioritizing query speed over storage
- Real-time or dashboard reporting where fast joins are critical
- Example: Amazon’s internal sales analytics dashboard

When to Use Snowflake Schema:

- When storage is a concern and data redundancy must be minimized
- Datasets with complex hierarchies or shared dimensions
- Used in data warehouses like Snowflake, Redshift with efficient query optimizers
- Example: Back-end data warehouse for a multinational e-commerce platform analysing data across countries and regions

Trade-Offs

Aspect	**Star Schema**	**Snowflake Schema**
Simplicity	Easier to design and query	More complex relationships
Redundancy	High	Low
Maintenance	Simpler	More granular updates possible
Performance	High (with proper indexing)	May vary based on DBMS