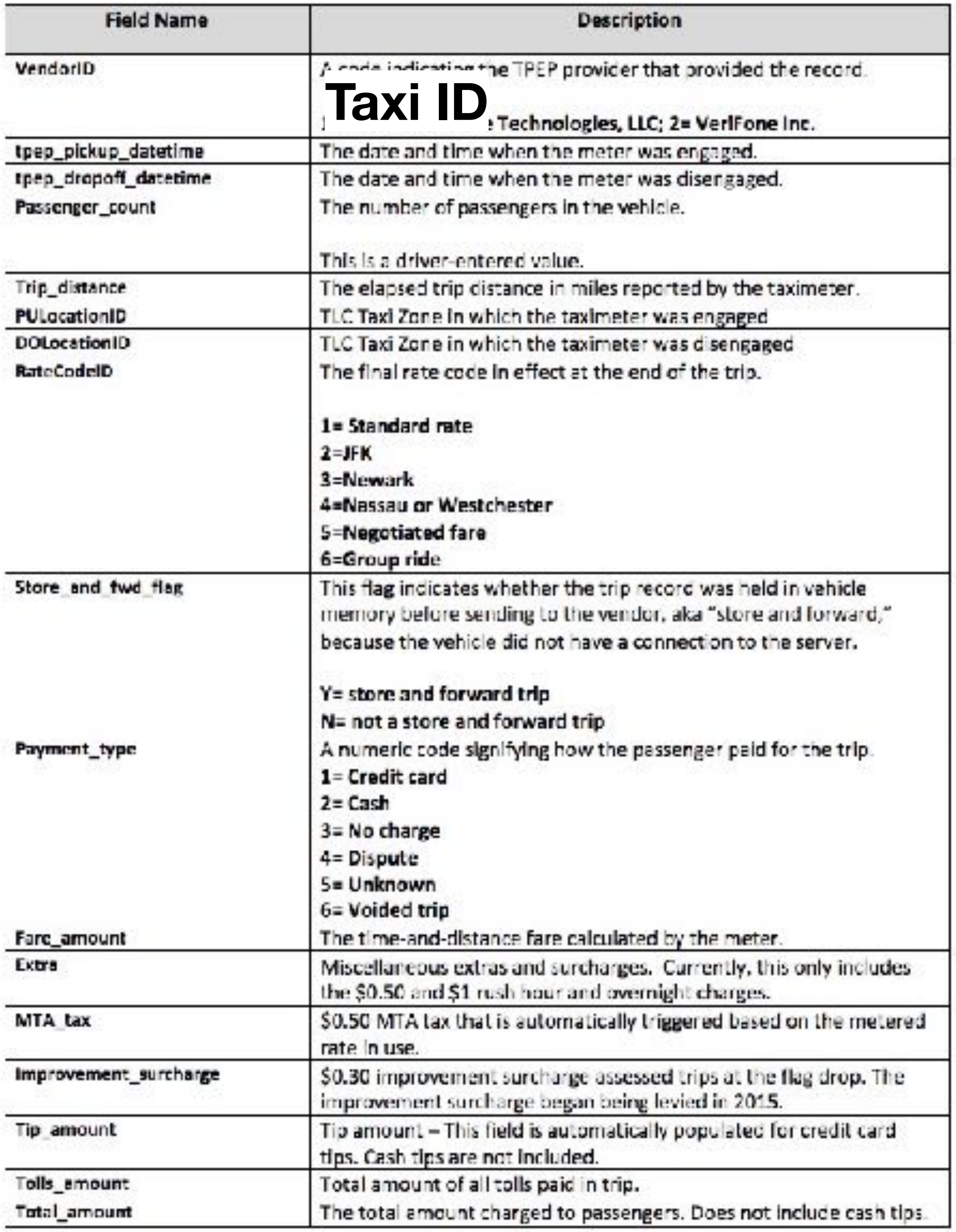
Data Models

Edmonton Data Management Meetup

Example …Taxi Business

- Simple business
  - Vehicle
  - Ride
  - Charges
  - Locations

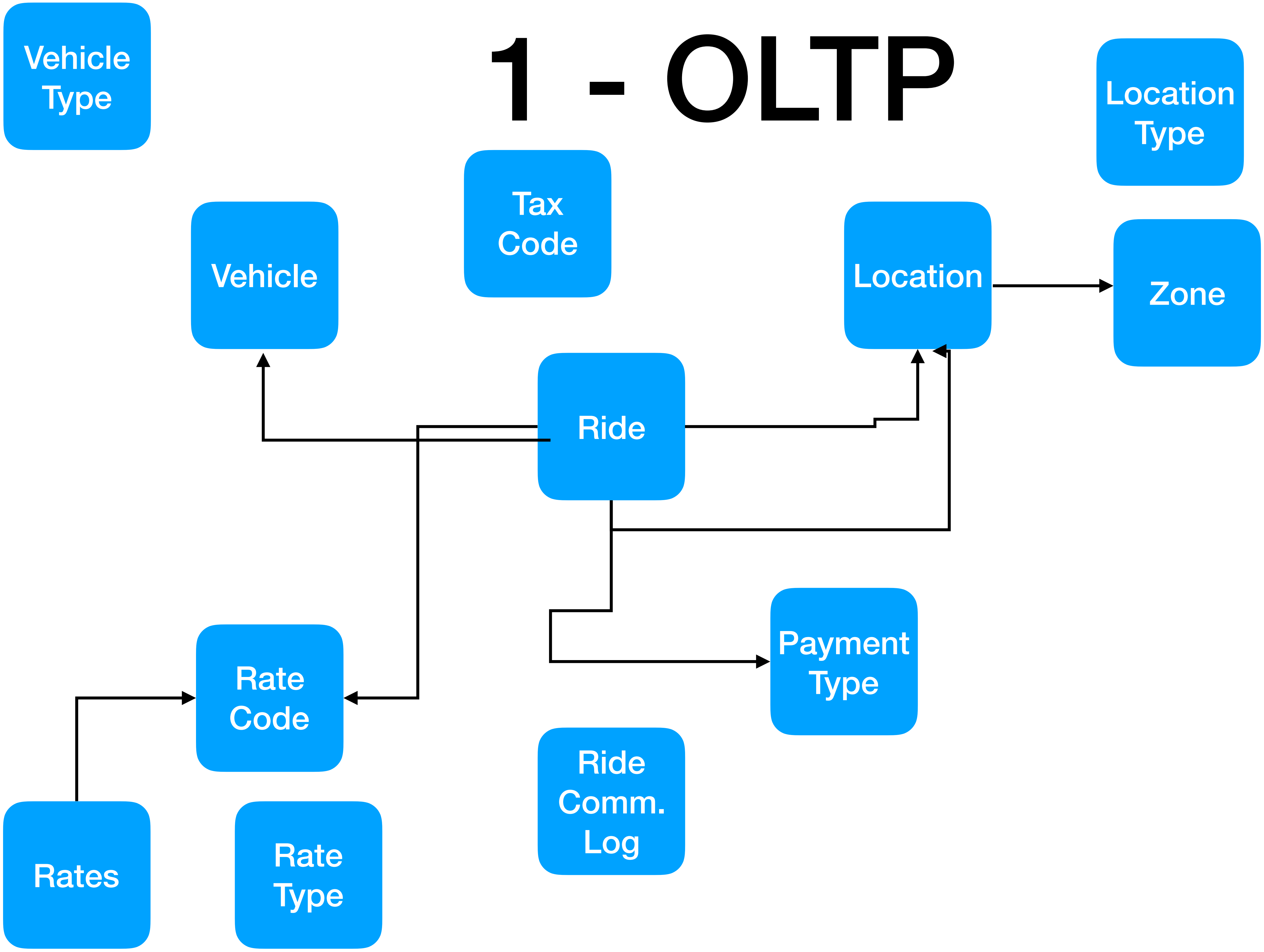


Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. Taxi ID 1 [illegible] Technologies, LLC; 2= VeriFone Inc.
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. **Y= store and forward trip** **N= not a store and forward trip**
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
MTA_tax	$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.

1 - OLTP

- Classic applications OLTP
- Method: 3NF (+)
- Tech: RDBMS
- Pros:
  - Great for data entry and maintainance
- Cons:
  - Tough reporting
  - Not Flexible

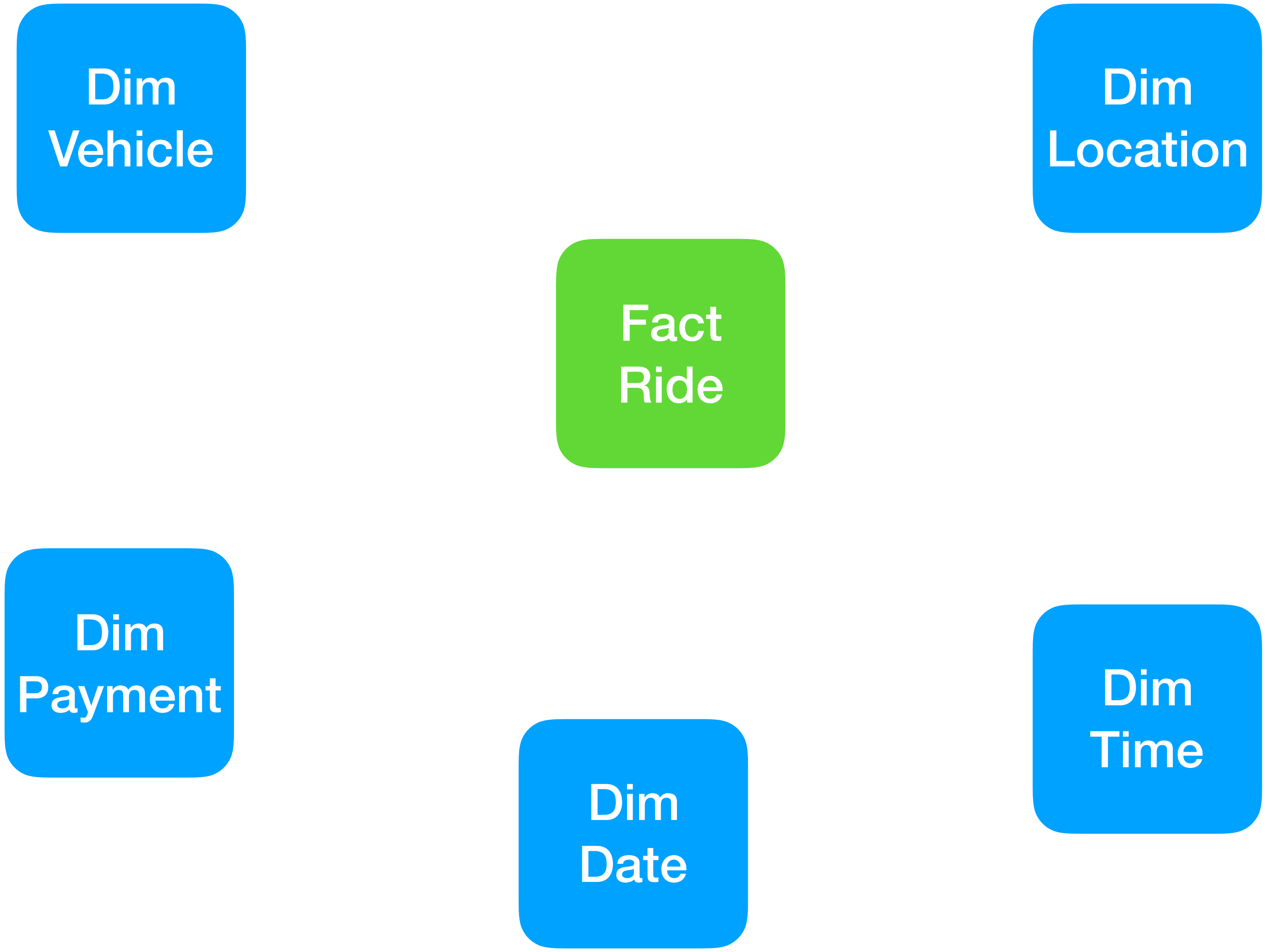
1 - OLTP


Vehicle Type
Tax Code
Location Type
Vehicle
Location
Zone
Ride
Payment Type
Rate Code
Ride Comm. Log
Rates
Rate Type

2 - Star Schema

- Used for Business Intelligence
- Method: Dimensional Models
- Tech: RDBMS
- Pros:
    - Simple Reporting
    - Very efficient
- Cons:
    - Not intended for updates

2 - Star Schema



Dim Vehicle

Dim Location

Fact Ride

Dim Payment

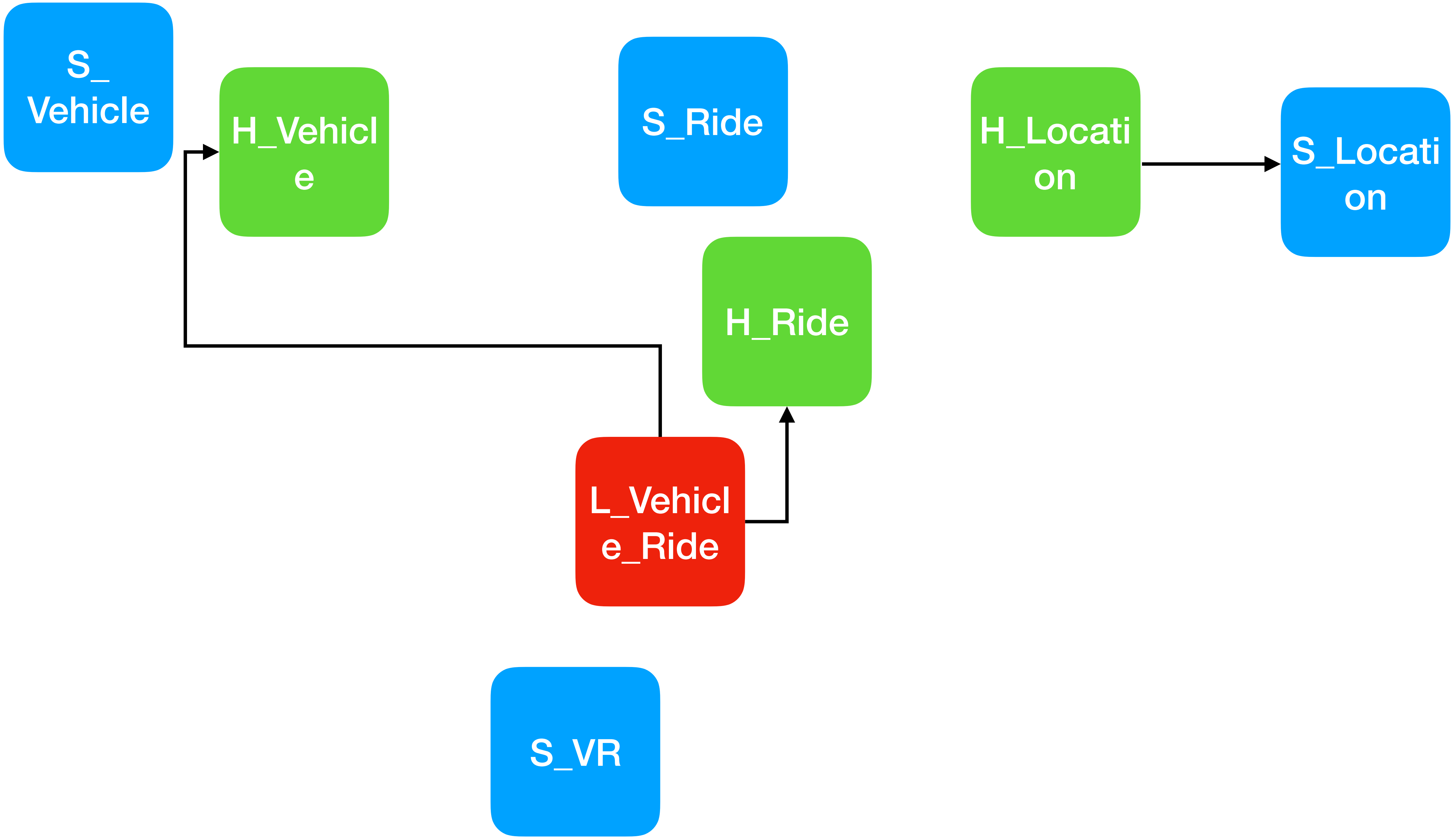
Dim Date

Dim Time

3 - Data Warehouse

- Used for Historical Storage and Tracking
- Method: Data Vault (Hub, Satellite, Link, +)
- Tech: RDBMS
- Pros:
  - Efficient
  - Flexible
- Cons:
  - More complicated than Dimensional to use

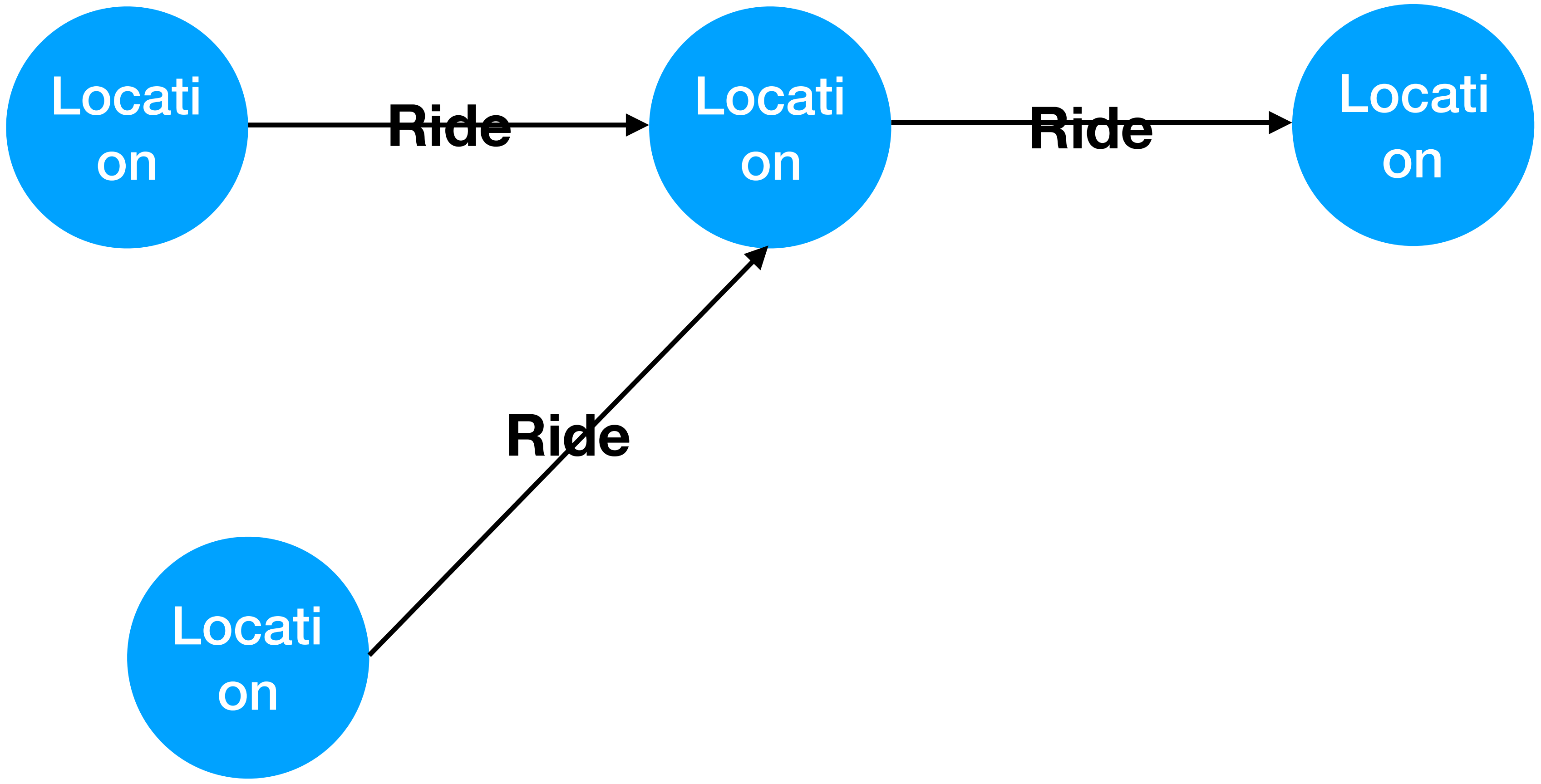
3 - Data Warehouse



4 - Graph

- Used for Explaining linkages
- Method: Property (Nodes & Edges) or RDF (Triples)
- Tech: Graph-centric DBs
- Pros:
    - Great at “Kevin Bacon”
- Cons:
    - Tough to load
    - Non standard SQL

4 - Graph



5 - Flat

- Old school - but still useful?
- Method: One big record
- Tech: Columnar stores or simple CSV
- Pros:
  - Simple to deal with
- Cons:
  - Require newer formats or tech to use

5 - Flat

VehicleID	PickupDTM	DropOffDTM	etc.

6 - JSON

- Latest version of structured data
- Method: JSON object
- Tech: Lots
- Pros:
  - Simple to deal with
  - Flexible
- Cons:
  - Programatic access or special tools
  - No real support for schemas

6 - JSON

```
• {
•         "Vehicle": 123,
•         "Times": {
•                 "Pickup": "2018-05-01 18:15",
•                 "DrofOff": "2018-05-01 19:05"
•         },
•         "Locations": {
•                 "Pickup": {
•                         "Address": "123 Broadway ST",
•                         "Burrugh": "Manhatten",
•                         "ZIP": "12345"
•                 },
•                 "DrofOff": {
•                         "Place": "Times Square"
•                 }
```

N - Probabilistic

- Used for Close Enough problems
- Method: Bloom Filters
- Tech: Special DBs
- Pros:
    - **very** quick and small
- Cons:
    - “No” or “Maybe”

N - Probabilistic

Probabilistic Data Structures

- Bloom Filter

- - Maybe or No
- - *TS* - identity TKS exist?

- HyperLogLog

- - Add, Count, or Merge
- - 12kb
- - Numbers are close enough
- - Venn diagrams

- **Count Min Sketch**

- - Frequency estimate
- - Hash add with weights

- **Cuckoo Filters**

- - Variation of Bloom Filter
- - Can delete and count
- - Better for read

Discussion