A APPENDIX

A.1 Accident Detection Using Speed Constraints

In autonomous driving, detecting accidents promptly is crucial for safety and efficient response. Sudden drops in speed, derived from changes in GPS coordinates over time, often signal potential collisions or abrupt braking events. Our approach employs speed constraints to validate incoming streams of GPS position data, where each record consists of x and y coordinates and timestamps ts. These constraints assess the rate of positional change over time, flagging any violation as a potential accident indicator.

The speed constraints are applied separately to the x and y directions, ensuring that changes in position over time fall within expected limits:

$$\forall x, y \in S \quad -2 \leq \frac{x.\mathtt{x} - y.\mathtt{x}}{x.\mathtt{ts} - y.\mathtt{ts}} \leq 2 \tag{IC1}$$

$$\forall x, y \in S \quad -2 \leq \frac{x.\mathtt{y} - y.\mathtt{y}}{x.\mathtt{ts} - y.\mathtt{ts}} \leq 2 \tag{IC2}$$

In data repair methods like [43], values outside these constraints might be corrected, which would erase critical evidence of an accident or an abrupt braking. Our solution, however, leave data untouched by annotating records with constraint violations rather than modifying them.

Let us consider the case of an autonomous-driving taxi service. A sample Continuous Query (CQ) below calculates the total distance traveled by each vehicle over a defined period (one day) and multiplies it by a fixed cost ratio for billing monitoring purposes.

```
SELECT vehicle_id, 1.5*(dist(x,y) / 1000) AS total_price
FROM GPSData [RANGE 1 DAY SLIDE 1 DAY]
GROUP BY vehicle_id;
```

Listing 3: A query calculating total bill by each vehicle in a given day, multiplying the total kilometers by a specific price of 1.5 dollars.

With speed constraints for constraint validation and CQ monitoring of total billing, our solution supports real-time data analysis from autonomous driving systems while considering potential malfunctions. Thanks to our approach, the malfunctions are not adjusted, they can be monitored in the query results, and manipulated depending on the scenario. For instance, the system can adjust the billing based on the presence of malfunctions. In such case, the exponent in the polynomial annotation can still represent the violation distance ϵ, and it can be used to summarize the entity of the malfunction. For instance, an abrupt braking that goes from 30 km/h to 0, may result in a deduction of 10% of the bill. An heavier violation, which corresponds to a more dangerous drop from 80 km/h to 0, may result in a much higher deduction, e.g., 60%.

A.2 Summary of Notations & Queries' Dataflow Representation

Table 5: Summary of used notation, □ is placeholder.

Concept	Notation
Stream	S
Record at time t in Stream S	$\tau_t(S)$
Record r with index i	r_i
Timestamp of Record r_i	$r_i.ts$
Window function	W
Interval	$\mathbb{W}$
Sub-Stream on Interval $\mathbb{W}$	$S[\mathbb{W}]$
Set of all Possible Intervals	$\mathcal{W}$
Window-less (whole stream) CQ	$\mathcal{Q}$
Window-based CQ	$\mathcal{Q}^W$
Window-less (whole stream) CQ on Constraint IC	$\mathcal{Q}_{IC}$
Window-based CQ on Constraint IC	$\mathcal{Q}^W_{IC}$
Result Stream of a (window-based) CQ on Stream S	$\mathcal{Q}^{\square}(S)$
$\mathbb{W}$-relation	$\mathcal{A}^{\mathbb{W}}$
Set of polynomials	$\mathbb{N}[\mathbb{X}]$
Set of Polynomials limited by $\mathbb{W}$	$\mathbb{N}[\mathbb{W}(\mathbb{X})]$
Violation Distance wrt r_k	ϵ_k

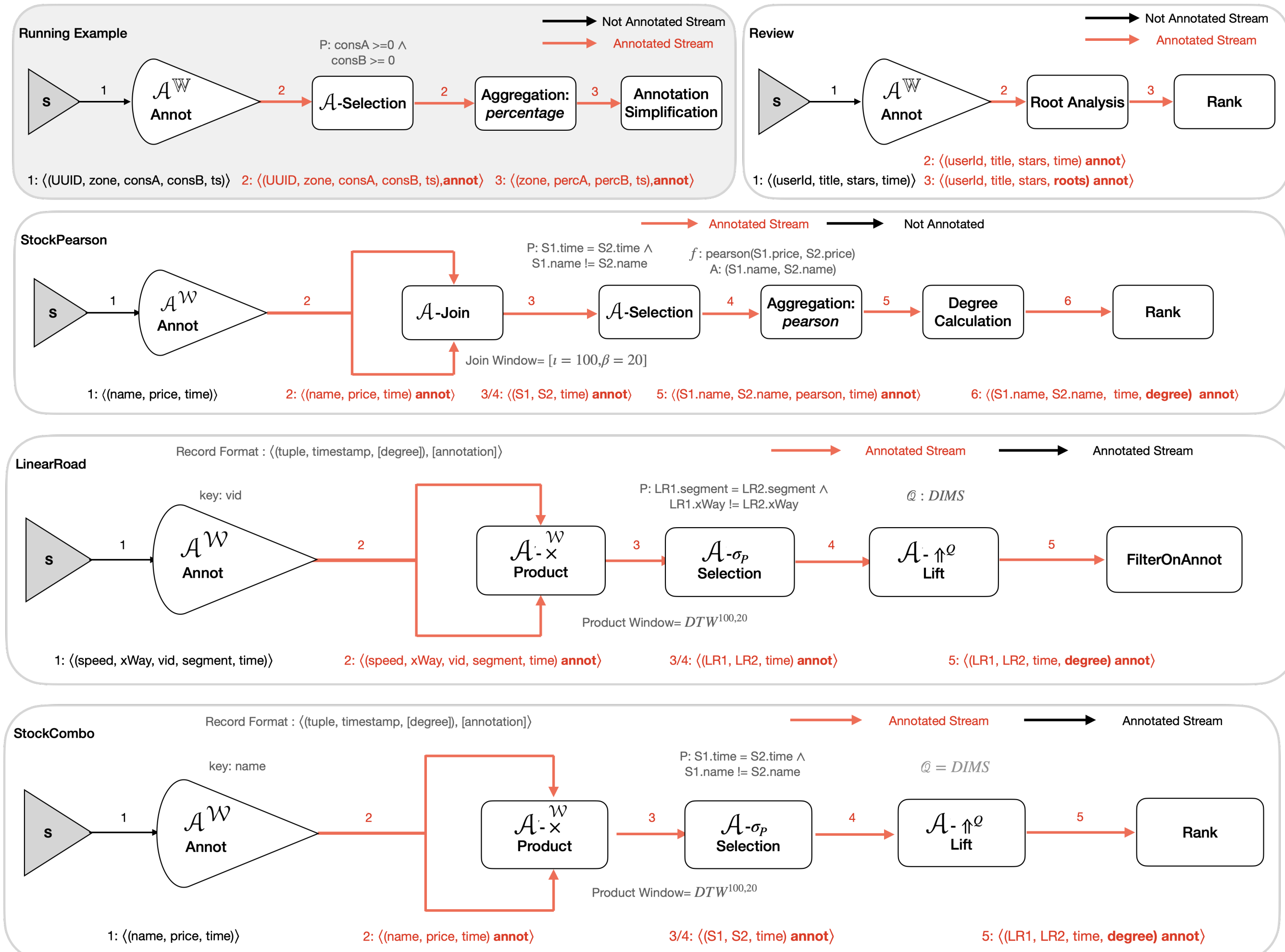

Figure 16: The queries used in the evaluation are represented in a dataflow-like format. We emphasize the annotated streams in red and display the schema of the intermediate streams.