**Goal of Orion**: To support uncertain data as a first class data type, and to provide querying and data storage utilities for uncertain data.

---

**Example Applications**

- Data integration and automatic cleaning
  - e.g. schema mapping, record linkage, etc.
- Information retrieval (keyword → structure)
  - e.g. deriving structure from keyword queries
- Spatio-temporal and sensor databases
  - e.g. measurement errors, outdated information
- Privacy preservation and obfuscation
  - e.g. dealing with statistics / anonymized values
- Scientific data management
  - e.g. assumptions in both raw and derived data

---

**Orion 2.0 Model**

- Uncertainty is handled by adding support for pdf attributes (i.e. `Uncertain` attributes)
- Standard distributions are stored in symbolic form if possible, or as approximations (e.g., histograms)
- Multiple attributes can be jointly distributed (intra-tuple dependencies)
- Inter-tuple dependencies are tracked using a directed, acyclic History Graph
- All database operations are expressed as three basic operations on pdfs: floor, product, marginalize

---

**Example Databases**

- **Example**: Sensor database

---

**Example Queries**

- **Marginalize**
  - SELECT xpos from sensor;
- **Floor**
  - SELECT * FROM sensor WHERE xpos > 330
- **Product**
  - CREATE TABLE temp_x AS SELECT xpos FROM sensor WHERE xpos > 330;
  - CREATE TABLE temp_y AS SELECT ypos FROM sensor WHERE ypos < 290;
  - SELECT xpos, ypos FROM temp_x, temp_y;

---

**Summary**

- First Model that handles continuous uncertain values (with PWS)
- Closed under basic database operations
- Unified Model that generalizes both attribute and tuple uncertainty (and more)
- Efficient and natural representation of data uncertainty
- Can handle both intra- and inter-tuple dependencies
- Implemented inside PostgreSQL