

Homework 1 Sample Answers

Schema:Sensor(sn, type, model)Station(name, lat, lon)Deployment(id, station, sensor, depth, startdate, sampling_factor)CTDObservation(deployment, time, conductivity, temperature, pressure)ADPObservation(deployment, time, depth, u, v)

```

CREATE VIEW CTDO AS (
  SELECT d.station, d.sensor, d.id as deployment
         d.sampling_factor as delta_t,
         s.lat as x, s.lon as y, o.depth as z, o.time as t,
         salinity(o.conductivity) as salinity,
         o.temperature,
         o.pressure
  FROM CTDObservation o, Deployment d, Station s
  WHERE o.deployment = d.id
        AND d.station = s.name
)

```

```

CREATE VIEW ADPO AS (
  SELECT d.station, d.sensor, d.id as deployment
         s.lat as x, s.lon as y, o.depth as z, o.time as t,
         o.u, o.v
  FROM ADPObservation o, Deployment d, Station s
  WHERE o.deployment = d.id
        AND d.station = s.name
)

```

Design:

CTD and ADP data are stored in separate tables, since 1) ADPs produce vectors and CTDs produce scalars, and 2) keeping them separate might simplify data loading.

The Deployment table allows sensors to be moved from station to station without losing track of the (x,y) coordinates of previous observations.

CS410/510: Scientific Data Management

We create two views to insulate applications from the specifics of our design; they do not need to write lengthy joins to access, say, the station name associated with an observation.

Q1:

```
SELECT r.name, t.name
   FROM Station r, Station t
  WHERE r.name = 'red26'
     AND t.name = 'tansy'
```

Q2:

```
SELECT r.temperature - t.temperature
   FROM CTDO r, CTDO t
  WHERE r.salinity = 0
     AND t.salinity = 0
     AND r.station = 'red26'
     AND t.station = 'tansy'
     AND ABS(r.t - t.t) < min(r.delta_t, t.delta_t)
```

Other interpretations of the join on time are possible.

Q3:

```
SELECT MAX(ABS(r.temperature - s.temperature))
   FROM CTDO r, CTDO s
  WHERE InRegion(r.x, r.y, xmax, xmin, ymax, ymin)
     AND InRegion(s.x, s.y, xmax, xmin, ymax, ymin)
     AND ABS(r.t - s.t) < min(r.delta_t, t.delta_t)
```

Q4:

If there's only one ADP and one CTD at red26:

```
SELECT c.t, c.temperature, v.t, AVG(v.u), AVG(v.v)
   FROM CTDO c, ADPO v
  WHERE c.station = 'red26'
     AND v.station = 'red26'
     AND ABS(c.t - v.t) < min(c.delta_t, v.delta_t)
 GROUP BY c.t, c.temperature, v.t
```

CS410/510: Scientific Data Management

Otherwise, there is room for interpretation. You could report an average over depth for each unique pair of CTD/ADP readings that are nearby in time:

```
SELECT c.sensor, c.t, c.temperature,
       v.sensor, v.t, AVG(v.u), AVG(v.v)
FROM CTDO c, ADPO v
WHERE c.station = 'red26'
      AND v.station = 'red26'
      AND ABS(c.t - v.t) < min(c.delta_t, v.delta_t)
GROUP BY c.sensor, c.t, v.sensor, v.t
```

Or you might report an average over all sensors for each, say, minute:

```
SELECT minute(c.t), AVG(c.temperature), AVG(v.u), AVG(v.v)
FROM CTDO c, ADPO v
WHERE c.station = 'red26'
      AND v.station = 'red26'
      AND minute(c.t) = minute(v.t)
GROUP BY minute(c.t)
```

Other interpretations are possible.

Q5:

Apologies for the typo: “delta_t” should be “delta_u, delta_v”.

```
SELECT t1.z as depth, t1.t as time,
       t1.u - t0.u as delta_u, t1.v - t0.v as delta_v,
FROM ADPO t0, ADPO t1
WHERE t0.z = t1.z
      AND t0.deployment = t1.deployment
      AND t1.t = (SELECT MIN(v.t)
                  FROM ADPO v
                  WHERE v.t > t0.t
                  AND v.deployment = t1.deployment
                  AND v.z = t1.z)
```

Other interpretations are possible.