

SQL: Part II

CPS 116  
Introduction to Database Systems

---

---

---

---

---

---

---

---

Announcements (September 18)

- ◆ Homework #1 due today (11:59pm)
  - Submit in class, slide underneath my office door
  - Sample solution available Thursday
- ◆ Homework #2 assigned today
  - Due next Thursday
- ◆ Project milestone #1 due in 3½ weeks

---

---

---

---

---

---

---

---

Incomplete information

- ◆ Example: *Student* (SID, name, age, GPA)
- ◆ Value unknown
  - We do not know Nelson's age
- ◆ Value not applicable
  - Nelson has not taken any classes yet; what is his GPA?

---

---

---

---

---

---

---

---

## Solution 1

### ❖ A dedicated special value for each domain (type)

- GPA cannot be  $-1$ , so use  $-1$  as a special value to indicate a missing or invalid GPA
- Leads to incorrect answers if not careful
  - `SELECT AVG(GPA) FROM Student;`
- Complicates applications
  - `SELECT AVG(GPA) FROM Student WHERE GPA <> -1;`
- Remember the Y2K bug?
  - "00" was used as a missing or invalid year value

---

---

---

---

---

---

---

---

## Solution 2

### ❖ A valid-bit for every column

- *Student* (*SID*, *name*, *name\_is\_valid*,  
*age*, *age\_is\_valid*,  
*GPA*, *GPA\_is\_valid*)
- Complicates schema and queries
  - `SELECT AVG(GPA) FROM Student WHERE GPA_is_valid;`

---

---

---

---

---

---

---

---

## Solution 3?

### ❖ Decompose the table; missing row = missing value

- *StudentName* (*SID*, *name*)
- *StudentAge* (*SID*, *age*)
- *StudentGPA* (*SID*, *GPA*)
- *StudentID* (*SID*)
- Conceptually the cleanest solution
- Still complicates schema and queries

---

---

---

---

---

---

---

---

## SQL's solution

- ❖ A special value NULL
  - For every domain
  - Special rules for dealing with NULL's
- ❖ Example: *Student* (*SID*, *name*, *age*, *GPA*)
  - ( 789, "Nelson", NULL, NULL )

---

---

---

---

---

---

---

---

## Computing with NULL's

- ❖ When we operate on a NULL and another value (including another NULL) using +, -, etc., the result is NULL
- ❖ Aggregate functions ignore NULL, except COUNT(\*) (since it counts rows)

---

---

---

---

---

---

---

---

## Three-valued logic

- ❖ TRUE = 1, FALSE = 0, UNKNOWN = 0.5
- ❖  $x$  AND  $y = \min(x, y)$
- ❖  $x$  OR  $y = \max(x, y)$
- ❖ NOT  $x = 1 - x$
- ❖ When we compare a NULL with another value (including another NULL) using =, >, etc., the result is UNKNOWN
- ❖ WHERE and HAVING clauses only select rows for output if the condition evaluates to TRUE
  - UNKNOWN is not enough

---

---

---

---

---

---

---

---