

SQL: Part II

CPS 116

Introduction to Database Systems

Announcements (September 20)

- ❖ Homework #2 assigned today
 - Due in 9 days (September 29)
- ❖ Homework #1 sample solution available
 - Hardcopies only
- ❖ Project Milestone #1 due in 23 days
 - Come to my office hours if you want to chat about project ideas

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;`

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`
- ❖ $x \text{ AND } y = \min(x, y)$
- ❖ $x \text{ OR } y = \max(x, y)$
- ❖ `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

Unfortunate consequences

- ❖ `SELECT AVG(GPA) FROM Student;`
`SELECT SUM(GPA)/COUNT(*) FROM Student;`
- ❖ `SELECT * FROM Student;`
`SELECT * FROM Student WHERE GPA = GPA;`
- ❖ Be careful: NULL breaks many equivalences
