DATABASE METHODOLOGY

Stockholm University

Relational Database Theory

The Relational Model Part 3 – NULL and Keys

The Relational Model – NULL and Keys

- In this module you will learn more about important properties of the Relational Model:
    - NULL – a problematic non-value
    - Keys and entity integrity
        - Superkeys
        - Candidate keys
        - Primary keys
        - Alternate keys
        - Surrogate keys

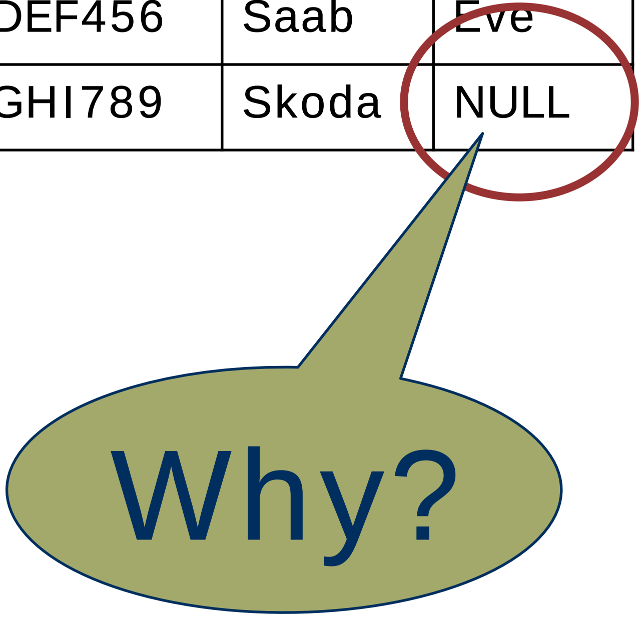
NULL – A Problematic Non-Value

- Attributes of relations (i.e. columns in tables) contain the data values in the Relational Model
- *But*, instead of “real” data values, attributes can contain a special “placeholder”, called **NULL**
    - NULL is not 0 (zero) or “” (the empty string)
    - NULL denotes that there *is no value*
    - NULL *cannot be compared* with anything
- NULLs can be *interpreted* in different ways (next slide)
- NULLs can be *eliminated* by careful database design

Interpretations Of NULL

- **Values do exist, but are currently unknown to the DB (database)**
  - The owner of car GHI789 might be registered at a *later time*
- **Values are simply not applicable to some tuples/rows in a table**
  - This could indicate a problem with the database design
- **Values are deliberately hidden by the DBA (database administrator)**
  - Authorized users may access them.
- **Worst case: We just don’t know!**

regNo	brand	owner
ABC123	Volvo	Peter
DEF456	Saab	Eve
GHI789	Skoda	NULL



Keys – The Tuple/Row Identifiers

- **Problem:** How to *unambigously* work with the *right* data in a relational database?
  - Every single data item must be addressable/searchable
    - Relations/tables have names – fine!
    - Attributes/columns have names – fine!
  - But how to access the right tuples/rows in a relation/table?
    - Tuples/rows have no names!

- **Solution:** Tuples/rows are unique (by definition):
  - So ,there must be at least one attribute/column (or a combination of attributes/columns) having unique values
    - Such attribute/column (or combination) is a **key**
    - Know the key value - find the right tuple/row!
    - There are similar but different kinds of key – next!

Superkeys And Candidate Keys

- **Superkey**
  - *Any combination* of attributes/columns (one or more, possibly all) whose values *uniquely identifies* a tuple/row in a relation/table
    - Can include attributes/columns that *aren't necessary* for the uniqueness
- **Candidate Key (CK)**
  - A *minimal* superkey
    - No proper subset of the CK is also a superkey
    - I.e. all attributes/columns in a CK *are necessary* for maintaining the uniqueness
  - All proper relations/tables have one (or more) CK

Primary Keys And Alternate Keys

- **Primary Key (PK)**
  - The CK that, during the database design, is chosen to identify the tuples/rows in a relation/table
    - There is <u>one and only one PK</u> in a relation/table
- **Alternate Key (AK)**
  - When the PK has been chosen, then any remaining CK is referred to as an Alternate Key (AK)

Person (in table view)

<u>ssn</u>	name	weight
111111-1111	Ollie	81
111111-2222	Peter	59
....	....	
999999-9999	Lisah	63

Simple syntax denoting the PK in a relation/table: Underline the attribute(s)/column(s) that is (are) part of the PK

Textual notation
Person(<u>ssn</u>, name, weight)

Entity Integrity And PKs Vs. NULL

Stockholm University

Entity Integrity

PK attribute(s)/column(s) in a relation/-table **must never be NULL**

This rule is called **entity integrity**

The PK must always be able to uniquely identify the tuples/rows, so all PK values must always exist completely

Remember: NULL is not a value and cannot be compared to anything, thus NULLs make it impossible to decide uniqueness

Note: AKs may have NULL-able attributes/columns (but avoid if possible)!

Person

ssn	name	weight
111111-1111	Ollie	81
111111-2222	Peter	59
....	....	
999999-9999	Lisah	63

In Person, ssn has been chosen to be the PK, thus a value ***must always*** exist here for ***every*** tuple/row

Surrogate (Primary) Keys

A PK can sometimes be problematic:

- It might lose its uniqueness with time
- Composites complicate data handling
- Users may disagree on the best PK

Solution:

- Introduce a **Surrogate Key (SK)**
  - Can be seen as an “artifical” CK
  - Contains no real information
  - Users should not see SK values
  - Mostly used internally by the RDBMS(*) for
    - finding tuples/rows
    - referencing relations/tables
- The “natural” CK(s) should still be:
  - Identified and documented
  - Implemented, as AK(s)

Residency

name	fromDate	toDate
Olle	2000-08-28	2000-09-01
Petia	1999-09-01	2006-01-02
Petia	2004-05-06	2004-05-07

Residency’

resID	name	fromDate	toDate
1678	Olle	2000-08-28	2000-09-01
1111	Petia	1999-09-01	2006-01-02
0004	Petia	2004-05-06	2004-05-07

SK!

A ***Surrogate Key (SK)*** is an *artificial* identifier, generated by the DBMS and guaranteed to be unique.

*** (R)DBMS: (Relational) Database Management System**
(E.g. MS Access, Oracle, SQL Server, DB2, MySQL, MariaDB)

Keys - Summarized

- The Primary Key (PK) of a relation/table:
  - Consists of one or more attributes/columns
    - If more than one, then we say it is a composite (primary) key
  - Is a Candidate Key (CK) in the first place
    - Thus, it is minimal with respect to its attributes/columns
  - Is always unique with respect to its values
    - Thus, no part of it may ever be NULL!
      - This is called entity integrity
- There is <u>one and only one</u> PK in each relation/table
  - But there can be one or more additional AK(s)

Medverkande

Anders Thelemyr – Lärare

Lars In de Betou – Mediepedagog

Inspelat 2015-08-31
Institutionen för data- och systemvetenskap, DSV

Stockholms universitet