



Nulls, Nested Queries & Views

COMP23111 - Database Systems

Gareth Henshall
Lecturer in Computer Science

Another look at NULL

- in SQL an unspecified data value is called NULL
- it's a **state**, not a value:
 - a NULL INT is not 0
 - a NULL FLOAT is not 0.000
 - a NULL VARCHAR(25) is not "" or filled with spaces
 - a NULL date is not 00/00/0000
- you can set a value to NULL
- you can test if a value is NULL
- a PK can never be NULL

w3schools SQL

[Null Values](#)

Another look at NULL

- NULL is a state, not a data value
- some possible semantics of NULL:
 - data value unknown
 - data unavailable
 - data withheld
 - data not applicable
- this leads to 3-valued logic

VIPs

<u>name</u>	flat	landline	mobile
Codd	13	496 0999	07700 900001
Chen	NULL	NULL	07700 900002
Everest	NULL	NULL	NULL
Widenius	8	495 0091	07700 900008

VIPS.sql on Bb, try it

Three-valued logic in SQL

- we're used to logic where a question is either TRUE or FALSE
- in SQL, we also have UNKNOWN

AND

	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
NULL	UNKNOWN	FALSE	UNKNOWN

OR

	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
NULL	TRUE	UNKNOWN	UNKNOWN

NOT

TRUE	FALSE
FALSE	TRUE
UNKNOWN	UNKNOWN

A longer story,
for interest only

modern-sql

3-valued logic

Checking for NULL

- we can check if an attribute is NULL or not with **IS**, but we can't test an attribute value for **equality** with NULL
- `SELECT name FROM VIPs WHERE flat = NULL;`

→ empty result

- `SELECT name FROM VIPs WHERE flat IS NULL;`

→
Chen
Everest

VIPs

<u>name</u>	flat	landline	mobile
Codd	13	496 0999	07700 900001
Chen	NULL	NULL	07700 900002
Everest	NULL	NULL	NULL
Widenius	8	495 0091	07700 900008

Nested query

- a query within a query
- returns data for use in the main query
- example. Find all students who are older than their average age

```
SELECT name, age FROM Student
WHERE age >
    (SELECT AVG (age) FROM Student);
```

➔ Cox 48

Student

name	<u>stuid</u>	age	major	tutor
Smith	17	21	Biology	18
Yang	31	19	Maths	17
Yang	73	18	AI	16
Cox	75	48	Physics	18

MySQL TUTORIAL

[Subquery](#)

procedure.sql on Bb, try it

Stored procedure

- define your own commands, sometimes useful

```
CREATE TABLE People (  
  name varchar(20),  
  age INT);
```

```
INSERT INTO People  
  VALUES ('Jim', 23), ('Pam', 19);
```

```
-- now define procedure listPeople()
```

```
DELIMITER /
```

```
CREATE PROCEDURE listPeople()
```

```
BEGIN
```

```
  SELECT * FROM People ORDER BY age;
```


```
END /
```

```
DELIMITER ;
```

```
-- now call the procedure
```

```
call listPeople();
```

commands temporarily switch the SQL statement-end marker to /, so SQL doesn't think the ; in the procedure is the end of the command (awkward? yes!)



Pam	19
Jim	23

People

name	age
Jim	23
Pam	19

Views

- a view is a virtual table created as the result of a predefined query
- the virtual table can itself be queried (but not changed unless privileged user)
- uses:
 - provide different views on data to different user communities
 - protect sensitive data

```
CREATE VIEW viewName AS
  SELECT column1, column2...
  FROM tableName
  WHERE [condition];
```


w3schools SQL

[CREATE VIEW](#)

View example

```
CREATE VIEW hasFlat AS  
  SELECT name, flat, landline  
  FROM VIPs  
  WHERE flat IS NOT NULL;
```

```
SELECT * FROM hasFlat;
```



Codd	13	496 0999
Widenius	8	495 0091

VIPs

<u>name</u>	flat	landline	mobile
Codd	13	496 0999	07700 900001
Chen	NULL	NULL	07700 900002
Everest	NULL	NULL	NULL
Widenius	8	495 0091	07700 900008

- why? the VIPs table may be restricted, so only the view **hasFlat** is visible to non-privileged users

w3schools SQL

[CREATE VIEW](#)