



Null Values

Three-Valued Logic

- In `SELECT ... WHERE ...`, a comparison between two values can yield
 - `TRUE`
 - `FALSE`
 - `NULL`
- During query processing:
 - DB discards rows evaluating to false or null
 - Retains only rows evaluating to true
- Often yields surprising results

Basic Comparisons

Comparisons yield NULL if either operand is NULL.

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT *  
FROM Club  
WHERE CBudget = NULL
```

- “CBudget = NULL” evaluates to <NULL>
- No rows in result

NULL in Simple Conditions

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT CName  
FROM Club  
WHERE CBudget IS NULL
```

- Always a mistake to use = or <> with NULL
- Use **IS NULL** or **IS NOT NULL**

CName

DELTA

NULL in Simple Conditions

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT *  
FROM Club  
WHERE CBudget < CActual
```

- Note: <NULL> < 1200.00 evaluates to <NULL>

Result 1		Messages		
CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C4	SIGMA	SOCIAL	600.00	750.00

Negation of NULL

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT *  
FROM Club  
WHERE NOT (CBudget < CActual)
```

NOT(<NULL>) is
<NULL>

Result 1		Messages		
CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00

NULL in Math

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT *  
FROM Club  
WHERE CBudget + 100 < CActual
```

Note: <NULL> + 100 evaluates to <NULL>

Result 1		Messages		
CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C4	SIGMA	SOCIAL	600.00	750.00

NULL and Aggregate Functions

- Null values ignored
- Effects can be subtle
 - `COUNT(*)` may differ from `Count(Column)`
 - `SUM(Column1) + SUM(Column2)` may differ from `SUM(Column1 + Column2)`

COUNT(*) vs COUNT(CBUDGET)

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	<NULL>	1200.00
C2	BITS	ACADEMIC	500.00	350.00
C3	HELPS	SERVICE	300.00	300.00
C4	SIGMA	SOCIAL	600.00	750.00

```
SELECT COUNT (*)  
FROM Club  
WHERE CPURPOSE = 'SOCIAL'
```

→ 2

```
SELECT COUNT (CBUDGET)  
FROM Club  
WHERE CPURPOSE = 'SOCIAL'
```

→ 1
(NULL excluded)

Compound Conditions

AND	True	False	Null
True	True	False	Null
False	False	False	False
Null	Null	False	Null

OR	True	False	Null
True	True	True	True
False	True	False	Null
Null	True	Null	Null

NOT	True	False	Null
	False	True	Null

Compound Conditions

- AND, OR
 - You can generally think about NULL operands as behaving like FALSE
- NOT
 - The tricky one
 - NOT NULL is NULL

Compound Conditions

CLUBNO	CNAME	CPURPOSE	CBUDGET	CACTUAL
C1	DELTA	SOCIAL	null	1200
C2	BITS	ACADEMIC	500	350
C3	HELPS	SERVICE	300	300
C4	SIGMA	SOCIAL	800	750

- `SELECT * FROM Club
WHERE CPurpose = 'Social' AND CBudget > CActual`
- `SELECT * FROM Club
WHERE NOT(CPurpose = 'Social' AND CBudget > CActual)`