Oracle Database Design - 2024

KRYSTIAN WOJTKIEWICZ

Last Example from previous lecture

Task 7. *List the values of all attributes of the* `Functions` *relation*

```
SELECT * FROM Functions;
```

```
FUNCTION   MIN_MICE              MAX_MICE
---------- --------------------- -------------------
BOSS       90                    110
THUG       70                    90
CATCHING   60                    70
CATCHER    50                    60
CAT        40                    50
NICE       20                    30
DIVISIVE   45                    55
HONORARY   6                     25
```

8 rows selected

Example

Task 8. *Specify the functions that cats perform in each band.*

```
SELECT DISTINCT band_no,function FROM
Cats;
```

```
BAND_NO                FUNCTION
---------------------- ----------
1                      NICE
1                      DIVISIVE
3                      THUG
2                      CATCHING
1                      BOSS
2                      NICE
2                      CATCHER
4                      CATCHING
3                      CATCHING
2                      THUG
3                      CAT
3                      NICE
4                      CAT
4                      CATCHER
 14 rows selected
```

Remark

The simplest form of the SELECT command can be extended by:

- string, date or number constant (date or string in single quotes)

called pseudo-column.

Example

***Task 9.** Specify the minimum and maximum mice ration connected with each function*

```
SELECT function,'can eat from
',min_mice,
       ' to ',max_mice
FROM Functions;
```

```
FUNCTION   'CANEATFROM' MIN_MICE   'TO' MAX_MICE
---------- ------------- ---------- ---- --------
BOSS       can eat from  90          to  110
THUG       can eat from  70          to  90
CATCHING   can eat from  60          to  70
CATCHER    can eat from  50          to  60
CAT        can eat from  40          to  50
NICE       can eat from  20          to  30
DIVISIVE   can eat from  45          to  55
HONORARY   can eat from  6           to  25
```

8 rows selected

Example

***Task 10.** Specify the minimum and maximum mice ration connected with each function (use column aliases)*

```
SELECT function Role,'can eat
from ' " ",
       min_mice "Min mice", '
to ' " ",max_mice "Max mice"
FROM Functions;
```

```
ROLE                     Min mice        Max mice
---------- ------------- ---------- ---- --------
BOSS       can eat from  90          to  110
THUG       can eat from  70          to  90
CATCHING   can eat from  60          to  70
CATCHER    can eat from  50          to  60
CAT        can eat from  40          to  50
NICE       can eat from  20          to  30
DIVISIVE   can eat from  45          to  55
HONORARY   can eat from  6           to  25
```

8 rows selected

Example

Task 11. *Determine the annual mice consumption for each cat.*

```
SELECT name,
(NVL(mice_ration,0)+NVL(mice_extra,0))*12
       "Eats annually"
FROM Cats;
```

```
NAME            Eats annually
--------------- ----------------------
BARI            672
MICKA           864
LUCEK           516
SONIA           660
LATKA           480
DUDEK           480
MRUCZEK         1632
CHYTRY          600
KOREK           1056
BOLEK           1116
ZUZIA           780
RUDA            768
PUCEK           780
PUNIA           732
BELA            624
KSAWERY         612
JACEK           804
MELA            612
```

18 rows selected

Example

Task 12. *Specify the minimum and maximum mice ration connected with each function.*

```
SELECT function||' can eat
from '||min_mice||' to
'||max_mice||
       ' mice per month'
"Function possibilities"
FROM Functions;
```

```
Function possibilities
-----------------------------------------------------
BOSS can eat from 90 to 110 mice per month
THUG can eat from 70 to 90 mice per month
CATCHING can eat from 60 to 70 mice per month
CATCHER can eat from 50 to 60 mice per month
CAT can eat from 40 to 50 mice per month
NICE can eat from 20 to 30 mice per month
DIVISIVE can eat from 45 to 55 mice per month
HONORARY can eat from 6 to 25 mice per month
```

8 rows selected

Remark

The WHERE clause enables rows selection operations on the relation specified by the FROM clause.

This selection takes place according to the condition (value of the logical expression) placed after the WHERE clause.

Example

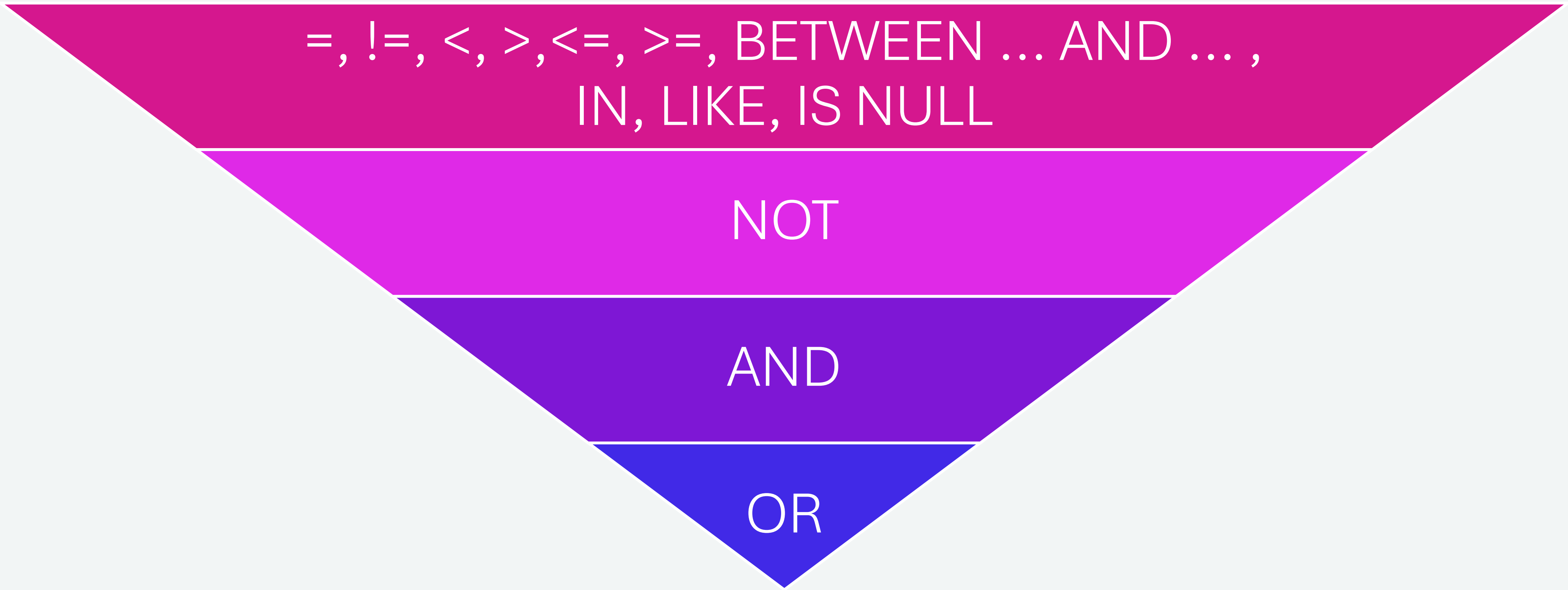
Task 13. *Find the names of all cats who perform the function NICE*

```
SELECT name

FROM Cats

WHERE function='NICE';
```

```
NAME

---------------

MICKA

SONIA

RUDA

BELA
```

4 rows selected

Remark



=, !=, <, >,<=, >=, BETWEEN … AND … ,
IN, LIKE, IS NULL

NOT

AND

OR

Remark

NULL value (in fact it is a lack of value!) is different from zero for numeric types or, e.g., from an empty sign for character types (in Oracle this value supported by NVL function).

It introduces, de facto, trivalent logic (TRUE, FALSE, NULL) instead of traditional divalent logic (TRUE, FALSE).

Remark

NOT	TRUE	FALSE	NULL
	FALSE	TRUE	NULL

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

Remark

NULL values may have attributes for which their optionality is allowed.

In this situation, the value of the arithmetic expression with the argument of NULL value is NULL.

An expression with a relational operator (e.g., <) that contains an arithmetic expression of NULL value also has a NULL value.

The NULL value in arithmetic expressions therefore requires special handling (e.g. via the NVL function).

Example

Task 14. *Find cats that don't get extra mice.*

```
SELECT nickname,gender||'   ' "GENDER"
FROM Cats
WHERE mice_extra IS NULL;
```

```
NICKNAME        GENDER
--------------- ------
TUBE            M
ZERO            M
EAR             W
SMALL           M
BOLEK           M
FAST            W
REEF            M
HEN             W
MAN             M
CAKE            M
LADY            W

11 rows selected
```

Example

Task 15. *Find cats with ration of mice between 50 and 60.*

```
SELECT nickname
FROM Cats
WHERE mice_ration BETWEEN 50
AND 60;
```

```
NICKNAME
----------------
TUBE
BOLEK
MAN
LADY
4 rows selected
```

Example

Task 16. *Find the names of cats performing the function of THUG or CATCHING, whose direct boss is TIGER.*

```
SELECT name

FROM Cats

WHERE function IN
('THUG','CATCHING') AND
chief='TIGER';
```

```
NAME

----------------

KOREK

BOLEK

PUCEK
```

3 rows selected

Example

Task 17. *Find cats whose have as a second letter O in their name.*

```
SELECT name

FROM Cats

WHERE name LIKE '_O%';
```

```
NAME

----------------

SONIA

KOREK

BOLEK
```

3 rows selected

Remark

The '_' character in the pattern means any character and the '%' character means any rest of the string. If it is necessary to check the presence of '_' or '%' in the string, they should be placed after the citation character defined in the ESCAPE clause, e.g .:

```
WHERE name LIKE '_&_U_&%U%' ESCAPE '&'
```

Example

Task 17. *Specify the nickname, function, ration of mice and ration extra for cats with not null ration extra of mice, whose ration of mice exceeds 70 or that have the function NICE.*

```
SELECT
function,NVL(mice_ration,0)
"MICE",mice_extra

FROM Cats

WHERE mice_extra IS NOT NULL

      AND

     (NVL(mice_ration,0)>70 OR
function='NICE');
```

```
FUNCTION   MICE          MICE_EXTRA
---------- ------------- ---------------
NICE       25            47
NICE       20            35
BOSS       103           33
THUG       75            13
THUG       72            21
NICE       22            42
NICE       24            28
```

7 rows selected

ORBER BY

The ORDER BY clause is used to explicitly order them against the value of the attribute/expression (list of attributes and/or expressions).

The following elements may appear in the clause:

- attribute identifier,
- expression,
- alias of the expression or of attribute from the SELECT clause,
- number of the expression/attribute in the SELECT clause.

The default ordering direction is ascending (ASC).

The descending direction is defined in ORDER BY clause by the word DESC after attribute/expression, by which ordering is performed.

Ordering can be also carried out implicitly. It is part of the following operations: CREATE INDEX, DISTINCT, GROUP BY, ORDER BY, INTERSECT, MINUS, UNION, joining of unindexed relations.

Example

***Task 18.** Display enemy data according decreasing hostility*

```
SELECT hostility_degree "How dangerous",
       enemy_name "Enemy name"
FROM Enemies
ORDER BY hostility_degree DESC;
```

```
How dangerous          Enemy name
---------------------- ---------------
10                     KAZIO
10                     WILD BILL
7                      UNRULY DYZIO
5                      SLYBOOTS
4                      DUN
3                      BASIL
2                      REKS
1                      BETHOVEN
1                      SLIM
1                      STUPID SOPHIA
```

10 rows selected

Example

Task 19. *Display data of cats for which the ration of mice exceeds 60. Sort data first ascending by gender and name of the band and then descending by date of join to the herd and then ascending by function name.*

```
SELECT nickname "Nickname",gender "Gender",

       band_no "Band",in_herd_since "Join
date",

       mice_ration "Eats"

FROM Cats WHERE mice_ration>60

ORDER BY 2,"Band",in_herd_since
DESC,function;
```

```
Nickname   Gender Band       Join date           Eats
---------- ------ ---------- ------------------- ----
TIGER      M      1          2002=01-01          103
CAKE       M      2          2008=12-01          67
BALD       M      2          2006=08-15          72
ZOMBIES    M      3          2004=03-16          75
REEF       M      4          2006=10-15          65
FAST       W      2          2006=07-21          65
HEN        W      3          2008=01-01          61
```

7 rows selected