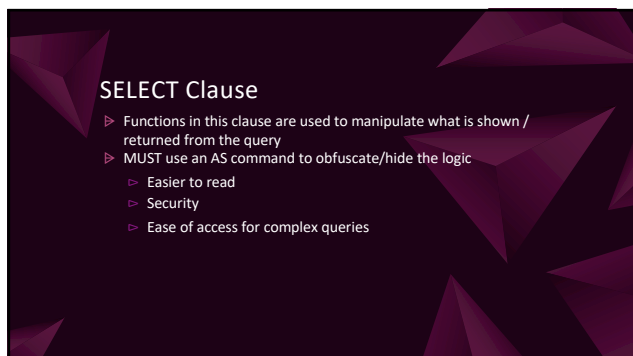
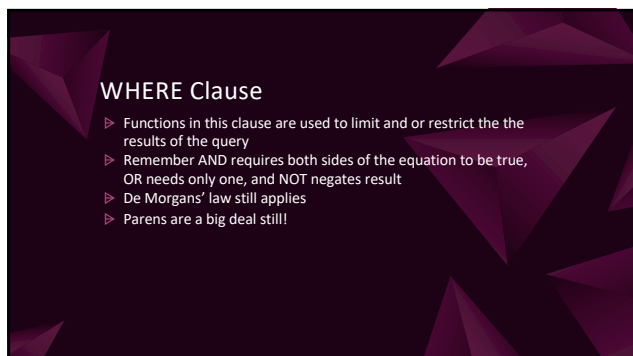


1



2



3

String Functions

- ▶ Manipulation and or extracting information from text
- ▶ String indices start at 1!!!!
- ▶ Info
 - ▶ LENGTH
- ▶ Padding (less used now)
 - ▶ LPAD, RPAD
- ▶ Trim
 - ▶ LTRIM, RTRIM, TRIM
- ▶ Partial
 - ▶ SUBSTR
- ▶ Manipulate
 - ▶ INSTR, CONCAT, REPLACE, UPPER, LOWER, INITCAP

4

String Functions Examples

Partial

```
SELECT
SUBSTR (first_name, 2) AS partial_first,
SUBSTR (last_name, LENGTH(last_name) / 2) AS last_half_name,
SUBSTR (first_name, -4) AS last_four_letters_of_first
FROM
  student;
```

Formatting

```
SELECT
LOWER(city) || ' ' || city AS small_and_normal_city,
UPPER(modified_by) AS BIG_MOD,
SUBSTR(city, 4) AS partial_city_name
FROM
  zipcode
WHERE
  MOD(zip, 2) = 0;
```

5

More String Examples

Padding

```
SELECT
LPAD(city, 50, 'x') AS left_pad_demo,
RPAD(city, 4, '?') AS right_pad_demo,
RPAD(city, 20, '?') AS right_pad_demo
FROM
  zipcode
WHERE
  zip < 5000;
```

Trimming

```
SELECT
LTRIM ('cody', 'c') AS left_trim,
RTRIM ('xxxxx 2', 'x') AS right_trim,
RTRIM ('xxxxxx yz', 'x') AS right_trim_two,
LTRIM ('cody') AS left_trim_two,
LTRIM ('xxxxxxx', 'x') AS l3
FROM
  dual;
```

6

Numeric Functions

- ▷ Only work on NUMERIC types
 - ▷ ABS (value)
 - ▷ MOD (value, divisor)
 - ▷ ROUND(value, precision)
 - ▷ TRUNC(value, precision)
 - ▷ FLOOR(value)
 - ▷ CEIL(value)

7

ROUND Example

```
SELECT
  ROUND (1234.127, 2) AS partial_round,
  ROUND (1274.127, -2) AS whole_round
FROM
  dual;
```

8

Arithmetic

- ▷ You can use the basic operators in your queries and they operate as you would expect

```
▷ ()
▷ *
▷ /
▷ +
▷ -
```

9

CASE statement

- How to handle multiple options in SQL
- Just like an if/else if/else block
- Order of evaluation matters!

CASE

- WHEN field comparison THEN value
- WHEN field other comparison THEN value
- ELSE default

END AS named_column

10

CASE Example

```
SELECT
  first_name,
  CASE
    WHEN zip < 1000 THEN 'here'
    WHEN zip < 3999 THEN 'there'
    WHEN zip > 4000 THEN 'over there ish'
    ELSE 'idk'
  END AS where_are_they
FROM
  student;
```

11

Handling NULL values

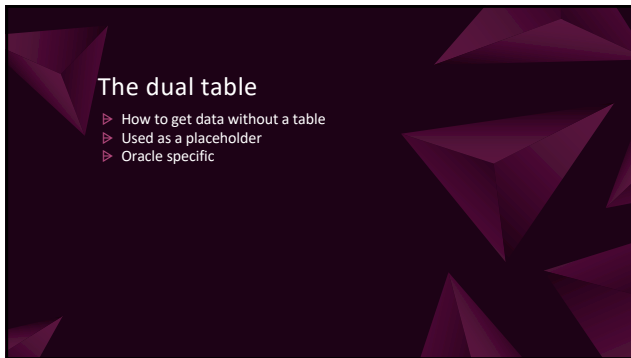
OK: NVL

```
SELECT
  NVL (field_name, '') AS data_source
FROM
  table_name;
```

Better: CASE

```
SELECT
  CASE
    WHEN field_name IS NULL THEN 'empty'
    WHEN field_name < 123 THEN 'small'
    ELSE 'BADAHA'
  END AS datasource
FROM
  table_name;
```

12



13
