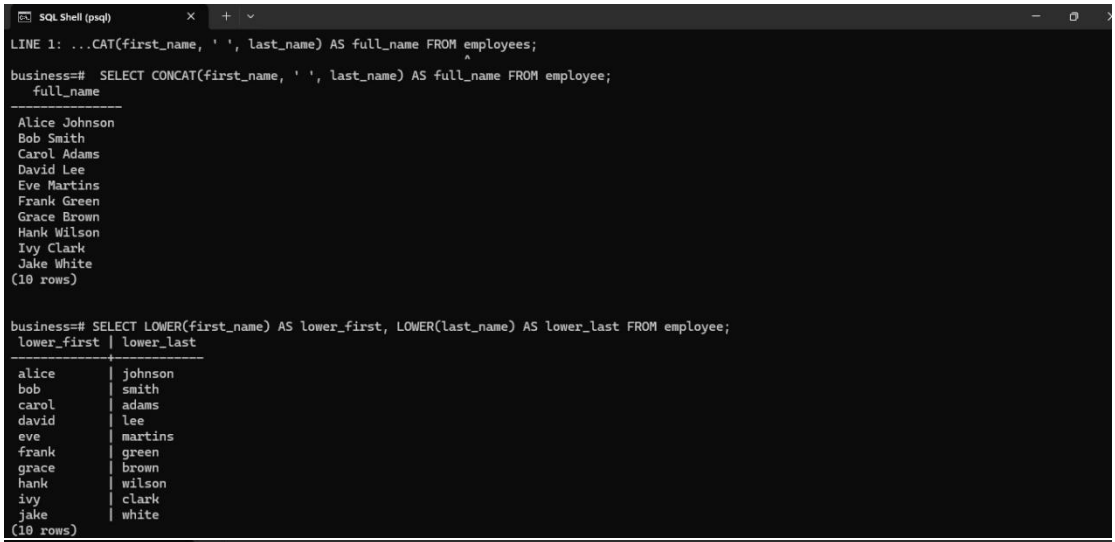
Names: abijuru jonathan
Id 29071

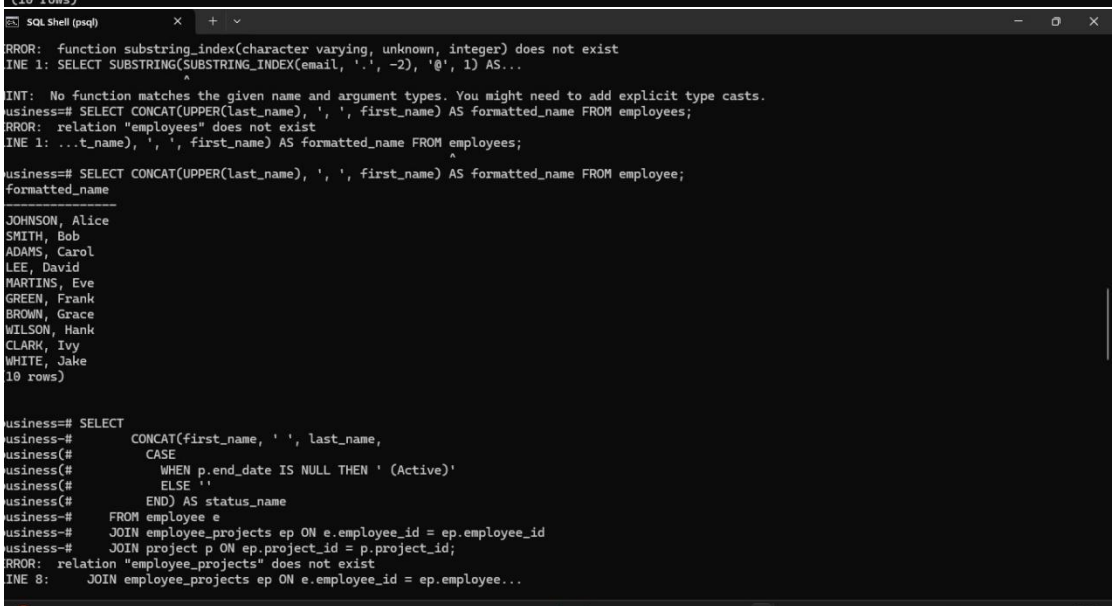
Note : some screenshot represent 2 queries

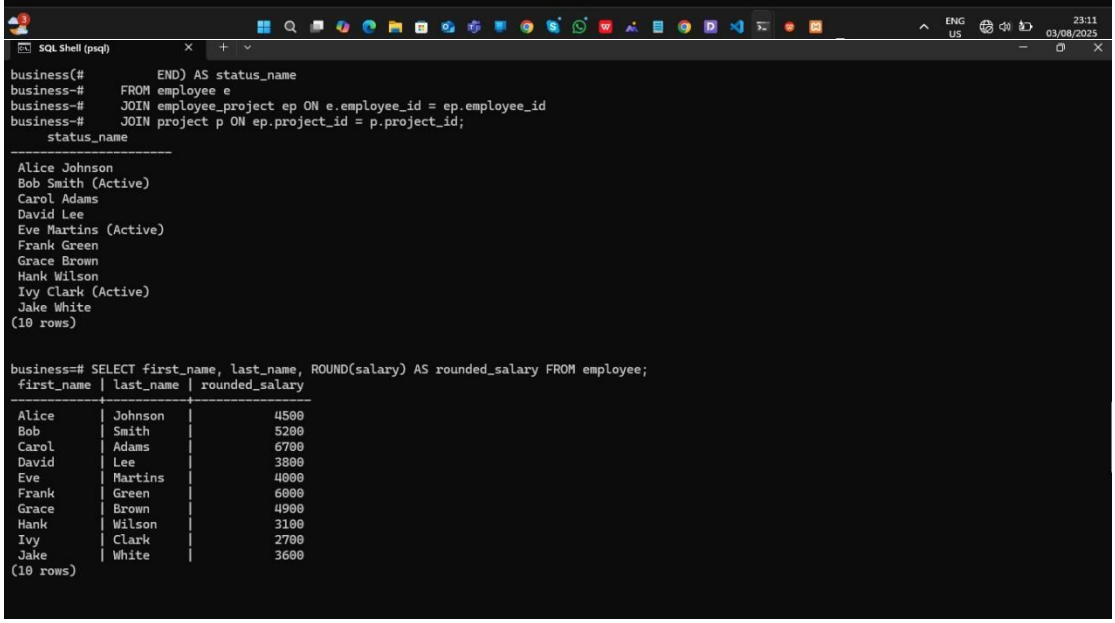
```
SQL Shell (psql)

business=# SELECT REPLACE(project_name, '-', '') AS cleaned_name FROM project;
       cleaned_name
---------------------------
 HR Revamp
 Finance Automation
 IT Infrastructure Upgrade
 Marketing Blitz 2025
 Legal Compliance
 Customer Portal
 Sales Booster
 R&D Pilot
 Procurement Tracker
 Operations Streamline
(10 rows)


business=# SELECT CONCAT('Emp: ', first_name, ' ', last_name, ' (', department_name, ')') AS employee_label
business-#     FROM employees
business-#     JOIN departments ON employees.department_id = departments.department_id;
ERROR:  relation "employees" does not exist
LINE 2:     FROM employees
                 ^
business=#   \d
            List of relations
 Schema |       Name       | Type  |  Owner
--------+------------------+-------+----------
 public | department       | table | postgres
 public | employee         | table | postgres
 public | employee_project | table | postgres
 public | project          | table | postgres
(4 rows)


business=#  SELECT CONCAT('Emp: ', first_name, ' ', last_name, ' (', department_name, ')') AS employee_label
```



```
business-#     FROM employee
business-#     JOIN department ON employee.department_id = department.department_id;
             employee_label
-----------------------------------------
 Emp: Alice Johnson (Human Resources)
 Emp: Carol Adams (Finance)
 Emp: Eve Martins (Information Technology)
 Emp: Bob Smith (Information Technology)
 Emp: David Lee (Marketing)
 Emp: Grace Brown (Legal)
 Emp: Hank Wilson (Operations)
 Emp: Jake White (Customer Service)
 Emp: Frank Green (Sales)
 Emp: Ivy Clark (Research and Development)
(10 rows)


business=# SELECT LENGTH(email) AS email_length FROM employee;
 email_length
--------------
           25
           21
           23
           21
           23
           23
           23
           23
           21
           22
(10 rows)


business=# SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(email, '.', -2), '@', 1) AS last_name_part FROM employee;
ERROR:  function substring_index(character varying, unknown, integer) does not exist
```


SQL Shell (psql)
LINE 1: ...CAT(first_name, ' ', last_name) AS full_name FROM employees;
business=# SELECT CONCAT(first_name, ' ', last_name) AS full_name FROM employee;
full_name
Alice Johnson
Bob Smith
Carol Adams
David Lee
Eve Martins
Frank Green
Grace Brown
Hank Wilson
Ivy Clark
Jake White
(10 rows)
business=# SELECT LOWER(first_name) AS lower_first, LOWER(last_name) AS lower_last FROM employee;
lower_first | lower_last
alice | johnson
bob | smith
carol | adams
david | lee
eve | martins
frank | green
grace | brown
hank | wilson
ivy | clark
jake | white
(10 rows)

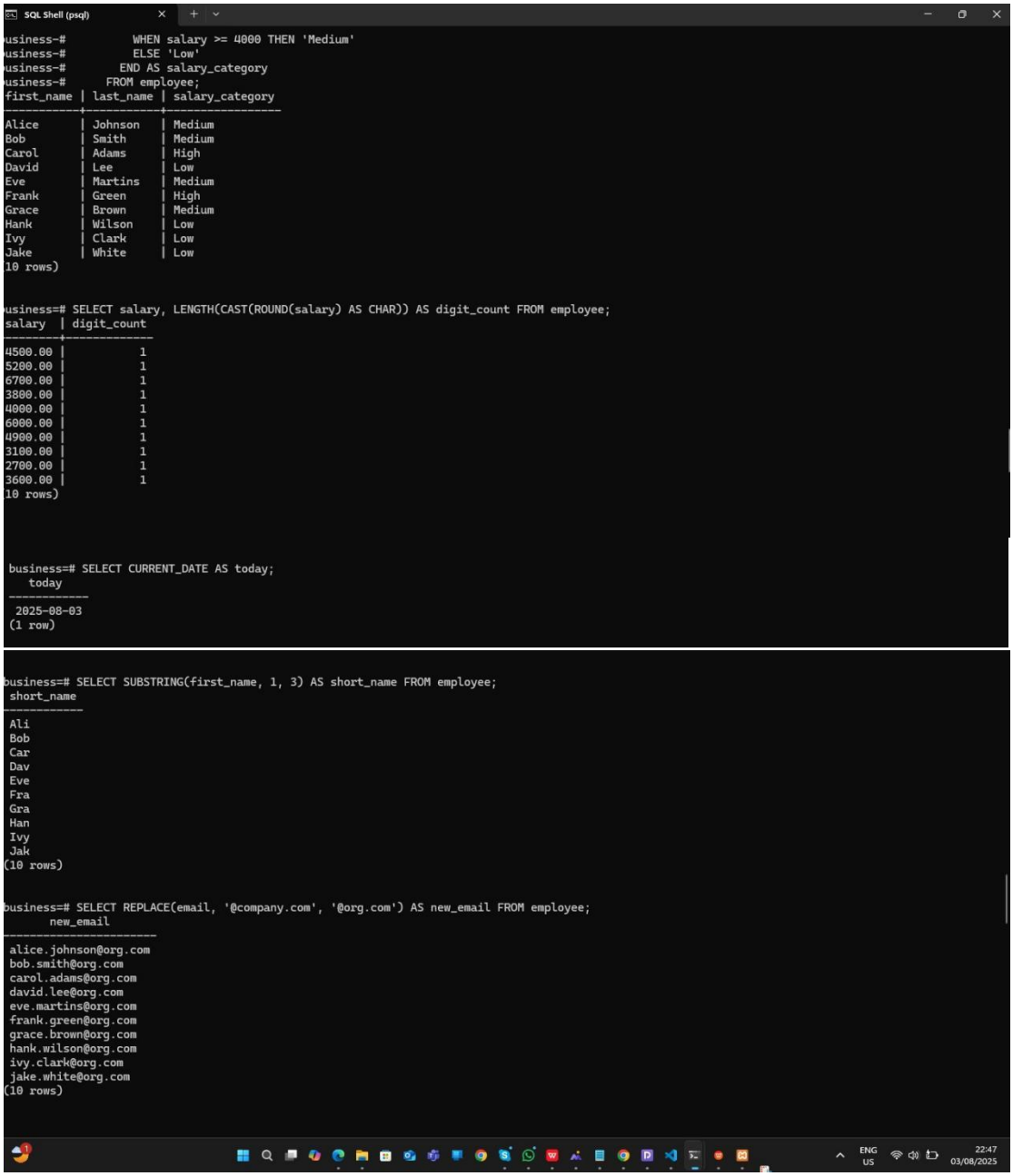

SQL Shell (psql)
ERROR: function substring_index(character varying, unknown, integer) does not exist
LINE 1: SELECT SUBSTRING(SUBSTRING_INDEX(email, '.', -2), '@', 1) AS...
HINT: No function matches the given name and argument types. You might need to add explicit type casts.
business=# SELECT CONCAT(UPPER(last_name), ', ', first_name) AS formatted_name FROM employees;
ERROR: relation "employees" does not exist
LINE 1: ...t_name), ', ', first_name) AS formatted_name FROM employees;
business=# SELECT CONCAT(UPPER(last_name), ', ', first_name) AS formatted_name FROM employee;
formatted_name
JOHNSON, Alice
SMITH, Bob
ADAMS, Carol
LEE, David
MARTINS, Eve
GREEN, Frank
BROWN, Grace
WILSON, Hank
CLARK, Ivy
WHITE, Jake
10 rows)
business=# SELECT
business-# CONCAT(first_name, ' ', last_name,
business(# CASE
business(# WHEN p.end_date IS NULL THEN ' (Active)'
business(# ELSE ''
business(# END) AS status_name
business-# FROM employee e
business-# JOIN employee_projects ep ON e.employee_id = ep.employee_id
business-# JOIN project p ON ep.project_id = p.project_id;
ERROR: relation "employee_projects" does not exist
LINE 8: JOIN employee_projects ep ON e.employee_id = ep.employee...


SQL Shell (psql)
business(# END) AS status_name
business-# FROM employee e
business-# JOIN employee_project ep ON e.employee_id = ep.employee_id
business-# JOIN project p ON ep.project_id = p.project_id;
status_name
Alice Johnson
Bob Smith (Active)
Carol Adams
David Lee
Eve Martins (Active)
Frank Green
Grace Brown
Hank Wilson
Ivy Clark (Active)
Jake White
(10 rows)
business=# SELECT first_name, last_name, ROUND(salary) AS rounded_salary FROM employee;
first_name | last_name | rounded_salary
Alice | Johnson | 4500
Bob | Smith | 5200
Carol | Adams | 6700
David | Lee | 3800
Eve | Martins | 4000
Frank | Green | 6000
Grace | Brown | 4900
Hank | Wilson | 3100
Ivy | Clark | 2700
Jake | White | 3600
(10 rows)

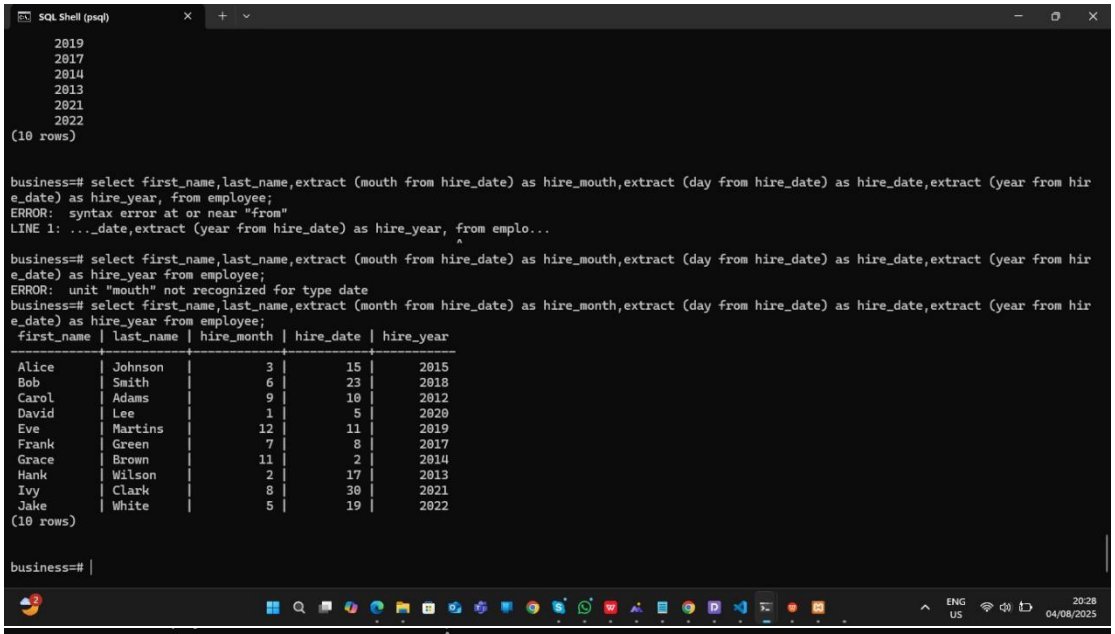
```
                ^
business=# SELECT first_name, last_name, salary
business-#      FROM employee
business-#     WHERE MOD(ROUND(salary), 2) = 0;
 first_name | last_name | salary
------------+-----------+---------
 Alice      | Johnson   | 4500.00
 Bob        | Smith     | 5200.00
 Carol      | Adams     | 6700.00
 David      | Lee       | 3800.00
 Eve        | Martins   | 4000.00
 Frank      | Green     | 6000.00
 Grace      | Brown     | 4900.00
 Hank       | Wilson    | 3100.00
 Ivy        | Clark     | 2700.00
 Jake       | White     | 3600.00
(10 rows)


business=# SELECT project_name, DATEDIFF(end_date, start_date) AS duration_days
business-#     FROM project
business-#     WHERE end_date IS NOT NULL;
ERROR:  function datediff(date, date) does not exist
LINE 1: SELECT project_name, DATEDIFF(end_date, start_date) AS durat...
                             ^
business-#     FROM employee e1, employee e2
business-#     WHERE e1.employee_id = 101 AND e2.employee_id = 103;
 salary_difference
-------------------
           2200.00
(1 row)


business=# SELECT first_name, last_name, salary, salary * POWER(1.10, 1) AS raised_salary FROM employee;
 first_name | last_name | salary  |      raised_salary
------------+-----------+---------+--------------------------
 Alice      | Johnson   | 4500.00 | 4950.000000000000000000
 Bob        | Smith     | 5200.00 | 5720.000000000000000000
 Carol      | Adams     | 6700.00 | 7370.000000000000000000
 David      | Lee       | 3800.00 | 4180.000000000000000000
 Eve        | Martins   | 4000.00 | 4400.000000000000000000
 Frank      | Green     | 6000.00 | 6600.000000000000000000
 Grace      | Brown     | 4900.00 | 5390.000000000000000000
 Hank       | Wilson    | 3100.00 | 3410.000000000000000000
 Ivy        | Clark     | 2700.00 | 2970.000000000000000000
 Jake       | White     | 3600.00 | 3960.000000000000000000
(10 rows)


business=#
```

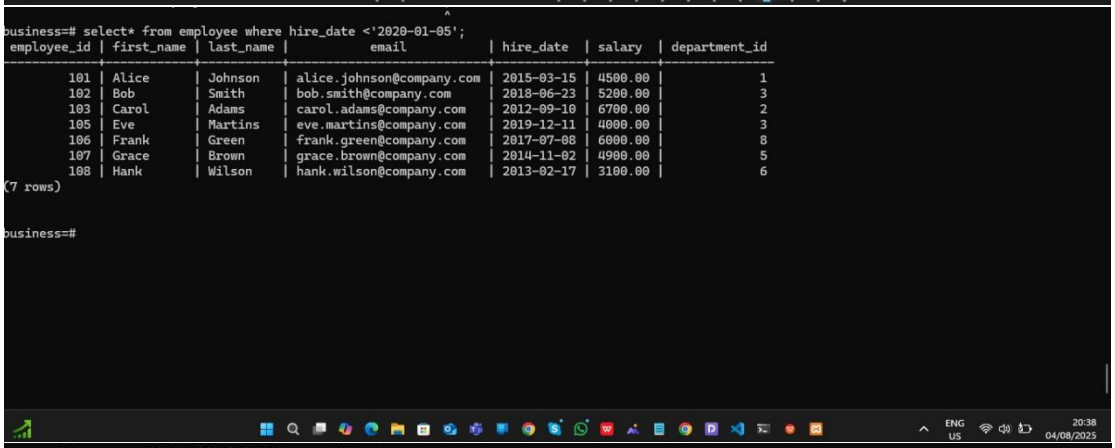
SQL Shell (psql)

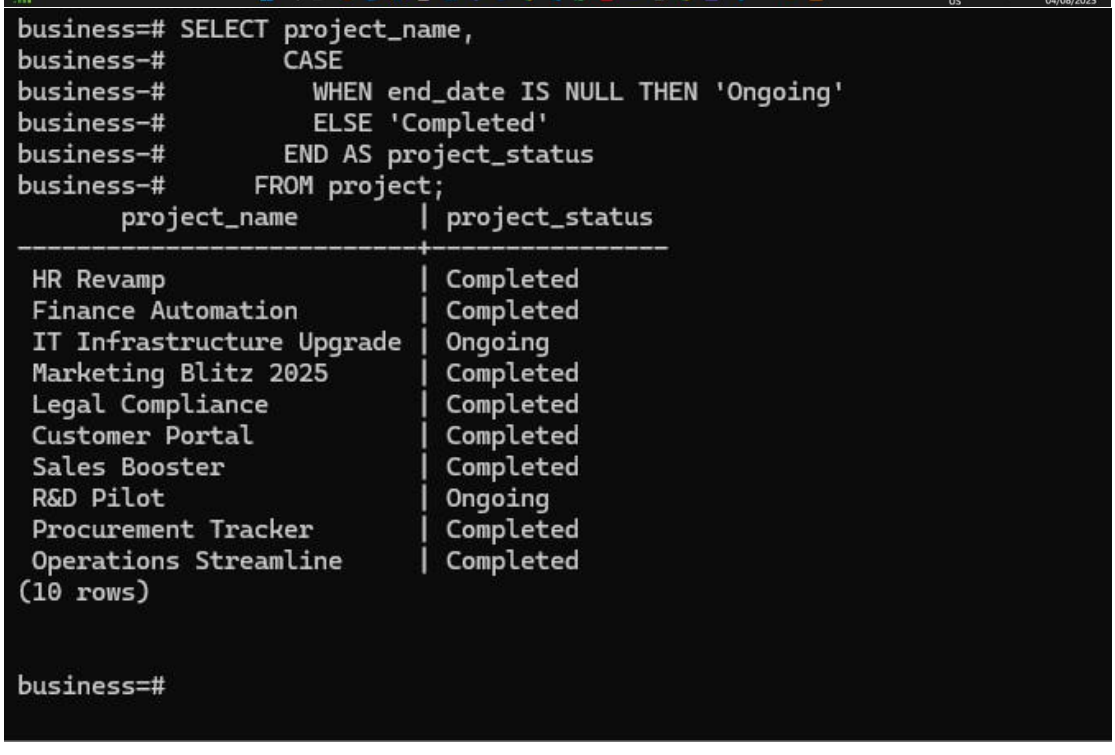
```
business=# SELECT salary, CEIL(salary) AS ceiling, FLOOR(salary) AS floor FROM employee;
 salary  | ceiling | floor
---------+---------+-------
 4500.00 |    4500 |  4500
 5200.00 |    5200 |  5200
 6700.00 |    6700 |  6700
 3800.00 |    3800 |  3800
 4000.00 |    4000 |  4000
 6000.00 |    6000 |  6000
 4900.00 |    4900 |  4900
 3100.00 |    3100 |  3100
 2700.00 |    2700 |  2700
 3600.00 |    3600 |  3600
(10 rows)


business=# SELECT first_name, last_name,
business-#        CASE
business-#          WHEN salary >= 6000 THEN 'High'
business-#          WHEN salary >= 4000 THEN 'Medium'
business-#          ELSE 'Low'
business-#        END AS salary_category
business-#     FROM employee;
 first_name | last_name | salary_category
------------+-----------+-----------------
 Alice      | Johnson   | Medium
 Bob        | Smith     | Medium
 Carol      | Adams     | High
 David      | Lee       | Low
 Eve        | Martins   | Medium
 Frank      | Green     | High
 Grace      | Brown     | Medium
 Hank       | Wilson    | Low
 Ivy        | Clark     | Low
 Jake       | White     | Low
```


SQL Shell (psql)
usiness-#   WHEN salary >= 4000 THEN 'Medium'
usiness-#   ELSE 'Low'
usiness-#   END AS salary_category
usiness-#   FROM employee;
first_name | last_name | salary_category
Alice | Johnson | Medium
Bob | Smith | Medium
Carol | Adams | High
David | Lee | Low
Eve | Martins | Medium
Frank | Green | High
Grace | Brown | Medium
Hank | Wilson | Low
Ivy | Clark | Low
Jake | White | Low
10 rows)
usiness=# SELECT salary, LENGTH(CAST(ROUND(salary) AS CHAR)) AS digit_count FROM employee;
salary | digit_count
4500.00 | 1
5200.00 | 1
6700.00 | 1
3800.00 | 1
4000.00 | 1
6000.00 | 1
4900.00 | 1
3100.00 | 1
2700.00 | 1
3600.00 | 1
10 rows)
business=# SELECT CURRENT_DATE AS today;
today
2025-08-03
(1 row)

business=# SELECT SUBSTRING(first_name, 1, 3) AS short_name FROM employee;
short_name
Ali
Bob
Car
Dav
Eve
Fra
Gra
Han
Ivy
Jak
(10 rows)
business=# SELECT REPLACE(email, '@company.com', '@org.com') AS new_email FROM employee;
new_email
alice.johnson@org.com
bob.smith@org.com
carol.adams@org.com
david.lee@org.com
eve.martins@org.com
frank.green@org.com
grace.brown@org.com
hank.wilson@org.com
ivy.clark@org.com
jake.white@org.com
(10 rows)
ENG
US
22:47
03/08/2025


SQL Shell (psql)
2019
2017
2014
2013
2021
2022
(10 rows)
business=# select first_name,last_name,extract (mouth from hire_date) as hire_mouth,extract (day from hire_date) as hire_date,extract (year from hir
e_date) as hire_year, from employee;
ERROR: syntax error at or near "from"
LINE 1: ..._date,extract (year from hire_date) as hire_year, from emplo...
^
business=# select first_name,last_name,extract (mouth from hire_date) as hire_mouth,extract (day from hire_date) as hire_date,extract (year from hir
e_date) as hire_year from employee;
ERROR: unit "mouth" not recognized for type date
business=# select first_name,last_name,extract (month from hire_date) as hire_month,extract (day from hire_date) as hire_date,extract (year from hir
e_date) as hire_year from employee;
first_name | last_name | hire_month | hire_date | hire_year
Alice | Johnson | 3 | 15 | 2015
Bob | Smith | 6 | 23 | 2018
Carol | Adams | 9 | 10 | 2012
David | Lee | 1 | 5 | 2020
Eve | Martins | 12 | 11 | 2019
Frank | Green | 7 | 8 | 2017
Grace | Brown | 11 | 2 | 2014
Hank | Wilson | 2 | 17 | 2013
Ivy | Clark | 8 | 30 | 2021
Jake | White | 5 | 19 | 2022
(10 rows)
business=#


^
business=# select* from employee where hire_date <'2020-01-05';
employee_id | first_name | last_name | email | hire_date | salary | department_id
101 | Alice | Johnson | alice.johnson@company.com | 2015-03-15 | 4500.00 | 1
102 | Bob | Smith | bob.smith@company.com | 2018-06-23 | 5200.00 | 3
103 | Carol | Adams | carol.adams@company.com | 2012-09-10 | 6700.00 | 2
105 | Eve | Martins | eve.martins@company.com | 2019-12-11 | 4000.00 | 3
106 | Frank | Green | frank.green@company.com | 2017-07-08 | 6000.00 | 8
107 | Grace | Brown | grace.brown@company.com | 2014-11-02 | 4900.00 | 5
108 | Hank | Wilson | hank.wilson@company.com | 2013-02-17 | 3100.00 | 6
(7 rows)
business=#


business=# SELECT project_name,
business-# CASE
business-# WHEN end_date IS NULL THEN 'Ongoing'
business-# ELSE 'Completed'
business-# END AS project_status
business-# FROM project;
project_name | project_status
HR Revamp | Completed
Finance Automation | Completed
IT Infrastructure Upgrade | Ongoing
Marketing Blitz 2025 | Completed
Legal Compliance | Completed
Customer Portal | Completed
Sales Booster | Completed
R&D Pilot | Ongoing
Procurement Tracker | Completed
Operations Streamline | Completed
(10 rows)
business=#

```
business-#          WHEN salary >= 4000 THEN 'Medium'
business-#          ELSE 'Low'
business-#        END AS salary_label
business-#      FROM employee;
 first_name | last_name | salary_label
------------+-----------+--------------
 Alice      | Johnson   | Medium
 Bob        | Smith     | Medium
 Carol      | Adams     | High
 David      | Lee       | Low
 Eve        | Martins   | Medium
 Frank      | Green     | High
 Grace      | Brown     | Medium
 Hank       | Wilson    | Low
 Ivy        | Clark     | Low
 Jake       | White     | Low
(10 rows)


business=#
```

```
 Bob        | Smith     | Medium
 Carol      | Adams     | High
 David      | Lee       | Low
 Eve        | Martins   | Medium
 Frank      | Green     | High
 Grace      | Brown     | Medium
 Hank       | Wilson    | Low
 Ivy        | Clark     | Low
 Jake       | White     | Low
(10 rows)


business=# SELECT first_name, last_name, COALESCE(email, 'No Email') AS
email_address
business-#      FROM employee;
 first_name | last_name |       email_address
------------+-----------+---------------------------
 Alice      | Johnson   | alice.johnson@company.com
 Bob        | Smith     | bob.smith@company.com
 Carol      | Adams     | carol.adams@company.com
 David      | Lee       | david.lee@company.com
 Eve        | Martins   | eve.martins@company.com
 Frank      | Green     | frank.green@company.com
 Grace      | Brown     | grace.brown@company.com
 Hank       | Wilson    | hank.wilson@company.com
 Ivy        | Clark     | ivy.clark@company.com
 Jake       | White     | jake.white@company.com
(10 rows)


business=#
```

```
business=#  SELECT first_name, last_name,
business-#        CASE
business-#          WHEN hire_date < '2015-01-01' THEN 'Veteran'
business-#          ELSE 'New'
business-#        END AS experience_status
business-#      FROM employee;
 first_name | last_name | experience_status
------------+-----------+-------------------
 Alice      | Johnson   | New
 Bob        | Smith     | New
 Carol      | Adams     | Veteran
 David      | Lee       | New
 Eve        | Martins   | New
 Frank      | Green     | New
 Grace      | Brown     | Veteran
 Hank       | Wilson    | Veteran
 Ivy        | Clark     | New
 Jake       | White     | New
(10 rows)


business=# +------------+-----
business-#
```

```
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c business;
You are now connected to database "business" as user "postgres".
business=# SELECT first_name, last_name, COALESCE(salary, 3000) AS salar
y_defaulted
business-#      FROM employee;
 first_name | last_name | salary_defaulted
------------+-----------+------------------
 Alice      | Johnson   |          4500.00
 Bob        | Smith     |          5200.00
 Carol      | Adams     |          6700.00
 David      | Lee       |          3800.00
 Eve        | Martins   |          4000.00
 Frank      | Green     |          6000.00
 Grace      | Brown     |          4900.00
 Hank       | Wilson    |          3100.00
 Ivy        | Clark     |          2700.00
 Jake       | White     |          3600.00
(10 rows)


business=#
```

```
 'IT'
business-#          WHEN department_name = 'Human Resources' THEN 'HR'
business-#          ELSE 'Other'
business-#        END AS department_category
business-#      FROM employee
business-#      JOIN department ON employee.department_id = department.d
epartment_id;
 first_name | last_name | department_category
------------+-----------+---------------------
 Alice      | Johnson   | HR
 Carol      | Adams     | Other
 Eve        | Martins   | IT
 Bob        | Smith     | IT
 David      | Lee       | Other
 Grace      | Brown     | Other
 Hank       | Wilson    | Other
 Jake       | White     | Other
 Frank      | Green     | Other
 Ivy        | Clark     | Other
(10 rows)
```

```
business=# SELECT e.first_name, e.last_name,
business-#        CASE
business-#          WHEN ep.employee_id IS NULL THEN 'Unassigned'
business-#          ELSE 'Assigned'
business-#        END AS assignment_status
business-#      FROM employee e
business-#      LEFT JOIN employee_project ep ON e.employee_id = ep.empl
oyee_id;
 first_name | last_name | assignment_status
------------+-----------+-------------------
 Alice      | Johnson   | Assigned
 Bob        | Smith     | Assigned
 Carol      | Adams     | Assigned
 David      | Lee       | Assigned
 Eve        | Martins   | Assigned
 Frank      | Green     | Assigned
 Grace      | Brown     | Assigned
 Hank       | Wilson    | Assigned
 Ivy        | Clark     | Assigned
 Jake       | White     | Assigned
(10 rows)


business=#
```

```
business=# SELECT first_name, last_name,
business-#        CASE
business-#          WHEN salary >= 6000 THEN '30% Tax Band'
business-#          WHEN salary >= 4000 THEN '20% Tax Band'
business-#          ELSE '10% Tax Band'
business-#       END AS tax_band
business-#     FROM employee;
 first_name | last_name |   tax_band
------------+-----------+--------------
 Alice      | Johnson   | 20% Tax Band
 Bob        | Smith     | 20% Tax Band
 Carol      | Adams     | 30% Tax Band
 David      | Lee       | 10% Tax Band
 Eve        | Martins   | 20% Tax Band
 Frank      | Green     | 30% Tax Band
 Grace      | Brown     | 20% Tax Band
 Hank       | Wilson    | 10% Tax Band
 Ivy        | Clark     | 10% Tax Band
 Jake       | White     | 10% Tax Band
(10 rows)
```

```
business=# SELECT employee_id, first_name,
business-#        CASE
business-#          WHEN MOD(employee_id, 2) = 0 THEN 'Even ID'
business-#          ELSE 'Odd ID'
business-#        END AS id_parity
business-#     FROM employee;
 employee_id | first_name | id_parity
-------------+------------+-----------
         101 | Alice      | Odd ID
         102 | Bob        | Even ID
         103 | Carol      | Odd ID
         104 | David      | Even ID
         105 | Eve        | Odd ID
         106 | Frank      | Even ID
         107 | Grace      | Odd ID
         108 | Hank       | Even ID
         109 | Ivy        | Odd ID
         110 | Jake       | Even ID
(10 rows)
```

```
business=# SELECT COALESCE(CONCAT(first_name, ' ', last_name), 'Unnamed
Employee') AS display_name from employee;
 display_name
---------------
 Alice Johnson
 Bob Smith
 Carol Adams
 David Lee
 Eve Martins
 Frank Green
 Grace Brown
 Hank Wilson
 Ivy Clark
 Jake White
(10 rows)


.   .     …


usiness=# SELECT email,
usiness-#        CASE
usiness-#          WHEN UPPER(email) LIKE '%TEST%' THEN 'Dummy Account'

usiness-#          ELSE 'Real Account'
usiness-#        END AS account_type
usiness-#      FROM employee;
          email           | account_type
--------------------------+--------------
alice.johnson@company.com | Real Account
bob.smith@company.com     | Real Account
carol.adams@company.com   | Real Account
david.lee@company.com     | Real Account
eve.martins@company.com   | Real Account
frank.green@company.com   | Real Account
grace.brown@company.com   | Real Account
hank.wilson@company.com   | Real Account
ivy.clark@company.com     | Real Account
jake.white@company.com    | Real Account
10 rows)


usiness=#
```

```
business=# SELECT first_name, last_name,
business-#        CASE
business-#          WHEN EXTRACT(month FROM hire_date) >= 2015 THEN 'Sen
ior'
business-#          WHEN EXTRACT(YEAR FROM hire_date) <= 2020 THEN 'Mid-
Level'
business-#          ELSE 'Junior'
business-#        END AS seniority
business-#      FROM employee;
 first_name | last_name | seniority
------------+-----------+-----------
 Alice      | Johnson   | Mid-Level
 Bob        | Smith     | Mid-Level
 Carol      | Adams     | Mid-Level
 David      | Lee       | Mid-Level
 Eve        | Martins   | Mid-Level
 Frank      | Green     | Mid-Level
 Grace      | Brown     | Mid-Level
 Hank       | Wilson    | Mid-Level
 Ivy        | Clark     | Junior
 Jake       | White     | Junior
(10 rows)


business=#
```

```
 ..                                                              ^
 business=# SELECT first_name, last_name,
 business-#        EXTRACT(MONTH FROM hire_date) AS hire_month,
 business-#        CASE
 business-#          WHEN EXTRACT(MONTH FROM hire_date) = EXTRACT(MONTH F
 ROM CURRENT_DATE)
 business-#          THEN 'Anniversary Month'
 business-#          ELSE 'Not Anniversary Month'
 business-#        END AS anniversary_status
 business-# FROM employee;
  first_name | last_name | hire_month |  anniversary_status
 ------------+-----------+------------+----------------------
  Alice      | Johnson   |          3 | Not Anniversary Month
  Bob        | Smith     |          6 | Not Anniversary Month
  Carol      | Adams     |          9 | Not Anniversary Month
  David      | Lee       |          1 | Not Anniversary Month
  Eve        | Martins   |         12 | Not Anniversary Month
  Frank      | Green     |          7 | Not Anniversary Month
  Grace      | Brown     |         11 | Not Anniversary Month
  Hank       | Wilson    |          2 | Not Anniversary Month
  Ivy        | Clark     |          8 | Anniversary Month
  Jake       | White     |          5 | Not Anniversary Month
 (10 rows)


 business=#
 business=#
```

```
business=# SELECT TRIM('   Revenue Authority   ') AS trimmed_text;
   trimmed_text
-------------------
 Revenue Authority
(1 row)


business=# SELECT LENGTH(CONCAT(first_name, last_name)) AS name_length FROM employee;
 name_length
-------------
          12
           8
          10
           8
          10
          10
          10
          10
           8
           9
(10 rows)


business=# SELECT INSTR(email, '@') AS at_position FROM employees;
ERROR:  relation "employees" does not exist
LINE 1: SELECT INSTR(email, '@') AS at_position FROM employees;
                                                     ^
```

SQL Shell (psql)

```
psql (17.5)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c business
You are now connected to database "business" as user "postgres".
business=# select upper(project_name) from project;
           upper
---------------------------
 HR REVAMP
 FINANCE AUTOMATION
 IT INFRASTRUCTURE UPGRADE
 MARKETING BLITZ 2025
 LEGAL COMPLIANCE
 CUSTOMER PORTAL
 SALES BOOSTER
 R&D PILOT
 PROCUREMENT TRACKER
 OPERATIONS STREAMLINE
(10 rows)


business=# SELECT REPLACE(project_name, '-', '') AS cleaned_name FROM project;
       cleaned_name
---------------------------
 HR Revamp
 Finance Automation
 IT Infrastructure Upgrade
 Marketing Blitz 2025
 Legal Compliance
 Customer Portal
 Sales Booster
```