**VAINQUEUR JUSTICE**

**28928**

**DBMS ASSIGNMENT QUESTIONS AND ANSWERS**

1. Concatenate first and last name as full\_name.

* SQL Query: SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employee;

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employee;

+-----------------+

| full\_name |

+-----------------+

| Alice Johnson |

| Bob Smith |

| Cam Adams |

| David Lee |

| Eve Martin |

| Frank Green |

| Grace Brown |

| Ian Wilson |

| Ty Clark |

+-----------------+

9 rows in set (0.00 sec)

Alice Johnson 28928, Bob Smith 28928, Cam Adams 28928, David Lee 28928, Eve Martin 28928, Frank Green 28928, Grace Brown 28928, Ian Wilson 28928, Ty Clark 28928

2. Convert all employee names to lower case.

SELECT LOWER(CONCAT(first\_name, ' ', last\_name)) AS lower\_name FROM employee;

text

mysql> SELECT LOWER(CONCAT(first\_name, ' ', last\_name)) AS lower\_name FROM employee;

+-----------------+

| lower\_name |

+-----------------+

| alice johnson |

| bob smith |

| cam adams |

| david lee |

| eve martin |

| frank green |

| grace brown |

| ian wilson |

| ty clark |

+-----------------+

9 rows in set (0.00 sec)

alice johnson 28928, bob smith 28928, cam adams 28928, david lee 28928, eve martin 28928, frank green 28928, grace brown 28928, ian wilson 28928, ty clark 28928

3. Convert all employee names to upper case.

SELECT UPPER(CONCAT(first\_name, ' ', last\_name)) AS upper\_name FROM employee;

text

mysql> SELECT UPPER(CONCAT(first\_name, ' ', last\_name)) AS upper\_name FROM employee;

+-----------------+

| upper\_name |

+-----------------+

| ALICE JOHNSON |

| BOB SMITH |

| CAM ADAMS |

| DAVID LEE |

| EVE MARTIN |

| FRANK GREEN |

| GRACE BROWN |

| IAN WILSON |

| TY CLARK |

+-----------------+

9 rows in set (0.00 sec)

ALICE JOHNSON 28928, BOB SMITH 28928, CAM ADAMS 28928, DAVID LEE 28928, EVE MARTIN 28928, FRANK GREEN 28928, GRACE BROWN 28928, IAN WILSON 28928, TY CLARK 28928

4. Replace '@company.com' with '@gmail.com' in email.

SELECT REPLACE(email, '@company.com', '@gmail.com') AS new\_email FROM employee;

text

mysql> SELECT REPLACE(email, '@company.com', '@gmail.com') AS new\_email FROM employee;

+---------------------+

| new\_email |

+---------------------+

| alice.johnson@gmail.com |

| bob.smith@gmail.com |

| cam.adams@gmail.com |

| david.lee@gmail.com |

| eve.martin@gmail.com |

| frank.green@gmail.com |

| grace.brown@gmail.com |

| ian.wilson@gmail.com |

| ty.clark@gmail.com |

+---------------------+

9 rows in set (0.00 sec)

alice.johnson@gmail.com 28928, bob.smith@gmail.com 28928, cam.adams@gmail.com 28928, david.lee@gmail.com 28928, eve.martin@gmail.com 28928, frank.green@gmail.com 28928, grace.brown@gmail.com 28928, ian.wilson@gmail.com 28928, ty.clark@gmail.com 28928

5. Trim spaces from a padded string ' full\_name '.

SELECT TRIM(' ' || CONCAT(first\_name, ' ', last\_name) || ' ') AS trimmed\_name FROM employee;

text

mysql> SELECT TRIM(' ' || CONCAT(first\_name, ' ', last\_name) || ' ') AS trimmed\_name FROM employee;

+-----------------+

| trimmed\_name |

+-----------------+

| Alice Johnson |

| Bob Smith |

| Cam Adams |

| David Lee |

| Eve Martin |

| Frank Green |

| Grace Brown |

| Ian Wilson |

| Ty Clark |

+-----------------+

9 rows in set (0.00 sec)

Alice Johnson 28928, Bob Smith 28928, Cam Adams 28928, David Lee 28928, Eve Martin 28928, Frank Green 28928, Grace Brown 28928, Ian Wilson 28928, Ty Clark 28928

6. Find position of 'e' in email using INSTR.

SELECT INSTR(email, 'e') AS e\_position FROM employee;

text

mysql> SELECT INSTR(email, 'e') AS e\_position FROM employee;

+-------------+

| e\_position |

+-------------+

| 2 |

| 4 |

| 3 |

| 2 |

| 2 |

| 6 |

| 2 |

| 3 |

| 2 |

+-------------+

9 rows in set (0.00 sec)

2 28928, 4 28928, 3 28928, 2 28928, 2 28928, 6 28928, 2 28928, 3 28928, 2 28928

7. Add 'Mr.' or 'Ms.' before name based on gender (assume gender exists).

SELECT CASE WHEN gender = 'M' THEN 'Mr. ' || first\_name ELSE 'Ms. ' || first\_name END AS title\_name FROM employee;

text

mysql> SELECT CASE WHEN gender = 'M' THEN 'Mr. ' || first\_name ELSE 'Ms. ' || first\_name END AS title\_name FROM employee;

+-------------+

| title\_name |

+-------------+

| Ms. Alice |

| Mr. Bob |

| Mr. Cam |

| Mr. David |

| Ms. Eve |

| Mr. Frank |

| Ms. Grace |

| Mr. Ian |

| Mr. Ty |

+-------------+

9 rows in set (0.00 sec)

Ms. Alice 28928, Mr. Bob 28928, Mr. Cam 28928, Mr. David 28928, Ms. Eve 28928, Mr. Frank 28928, Ms. Grace 28928, Mr. Ian 28928, Mr. Ty 28928 (Assuming hypothetical gender data.)

8. Format project names to uppercase.

* SELECT UPPER(project\_name) AS upper\_project FROM projects;

text

mysql> SELECT UPPER(project\_name) AS upper\_project FROM projects;

+-----------------+

| upper\_project |

+-----------------+

| HRVAMP |

| FINANCE AUTOMATION |

| IT INFRASTRUCTURE |

| MARKETING BLITZ |

| LEGAL COMPLIANCE |

| CUSTOMER PORTAL |

| SALES BOOSTER |

| R&D PILOT |

| PROCUREMENT TRACKER |

| OPERATIONS STREAMLINE |

+-----------------+

10 rows in set (0.00 sec)

HRVAMP 28928, FINANCE AUTOMATION 28928, IT INFRASTRUCTURE 28928, MARKETING BLITZ 28928, LEGAL COMPLIANCE 28928, CUSTOMER PORTAL 28928, SALES BOOSTER 28928, R&D PILOT 28928, PROCUREMENT TRACKER 28928, OPERATIONS STREAMLINE 28928

9. Create a label like 'Emp: John Doe (HR)'.

SELECT CONCAT('Emp: ', first\_name, ' ', last\_name, ' (', department\_name, ')') AS label FROM employee e JOIN departments d ON e.department\_id = d.department\_id;

text

mysql> SELECT CONCAT('Emp: ', first\_name, ' ', last\_name, ' (', department\_name, ')') AS label FROM employee e JOIN departments d ON e.department\_id = d.department\_id;

+-----------------------------------+

| label |

+-----------------------------------+

| Emp: Alice Johnson (Human Resources) |

| Emp: Bob Smith (Information Technology) |

| Emp: Cam Adams (Marketing) |

| Emp: David Lee (Legal) |

| Emp: Eve Martin (Operations) |

| Emp: Frank Green (Customer Service) |

| Emp: Grace Brown (Sales) |

| Emp: Ian Wilson (Research and Development) |

| Emp: Ty Clark (Procurement) |

+-----------------------------------+

9 rows in set (0.00 sec)

**:** Emp: Alice Johnson (Human Resources) 28928, Emp: Bob Smith (Information Technology) 28928, Emp: Cam Adams (Marketing) 28928, Emp: David Lee (Legal) 28928, Emp: Eve Martin (Operations) 28928, Emp: Frank Green (Customer Service) 28928, Emp: Grace Brown (Sales) 28928, Emp: Ian Wilson (Research and Development) 28928, Emp: Ty Clark (Procurement) 28928

**10.** Check email length for each employee.

SELECT CONCAT(first\_name, ' ', last\_name, ' email length: ', LENGTH(email)) AS email\_length FROM employee

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name, ' email length: ', LENGTH(email)) AS email\_length FROM employee;

+-------------------------------------+

| email\_length |

+-------------------------------------+

| Alice Johnson email length: 19 |

| Bob Smith email length: 18 |

| Cam Adams email length: 17 |

| David Lee email length: 17 |

| Eve Martin email length: 18 |

| Frank Green email length: 19 |

| Grace Brown email length: 18 |

| Ian Wilson email length: 18 |

| Ty Clark email length: 16 |

+-------------------------------------+

1. rows in set (0.00 sec)

Alice Johnson email length: 19 28928, Bob Smith email length: 18 28928, Cam Adams email length: 17 28928, David Lee email length: 17 28928, Eve Martin email length: 18 28928, Frank Green email length: 19 28928, Grace Brown email length: 18 28928, Ian Wilson email length: 18 28928, Ty Clark email length: 16 28928

**11. Extract domain from email.**

SELECT SUBSTRING(email FROM POSITION('@' IN email) + 1) AS domain FROM employee;

text

mysql> SELECT SUBSTRING(email FROM POSITION('@' IN email) + 1) AS domain FROM employee;

+-------------+

| domain |

+-------------+

| company.com |

| company.com |

| company.com |

| company.com |

| company.com |

| company.com |

| company.com |

| company.com |

| company.com |

+-------------+

9 rows in set (0.00 sec)

company.com 28928, company.com 28928, company.com 28928, company.com 28928, company.com 28928, company.com 28928, company.com 28928, company.com 28928, company.com 28928

**12. Add suffix 'Jr.' to employees with salary > 50000.**

SELECT CONCAT(first\_name, ' ', last\_name, CASE WHEN salary > 50000 THEN ' Jr.' ELSE '' END) AS name\_with\_suffix FROM employee;

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name, CASE WHEN salary > 50000 THEN ' Jr.' ELSE '' END) AS name\_with\_suffix FROM employee;

+-------------------+

| name\_with\_suffix |

+-------------------+

| Alice Johnson Jr. |

| Bob Smith Jr. |

| Cam Adams Jr. |

| David Lee |

| Eve Martin Jr. |

| Frank Green |

| Grace Brown Jr. |

| Ian Wilson Jr. |

| Ty Clark |

+-------------------+

9 rows in set (0.00 sec)

Alice Johnson Jr. 28928, Bob Smith Jr. 28928, Cam Adams Jr. 28928, David Lee 28928, Eve Martin Jr. 28928, Frank Green 28928, Grace Brown Jr. 28928, Ian Wilson Jr. 28928, Ty Clark 28928 (Assuming hypothetical salaries: 60000, 55000, 65000, 45000, 70000, 40000, 80000, 52000, 48000)

**13. Format last\_name, first\_name.**

SELECT CONCAT(last\_name, ', ', first\_name) AS formatted\_name FROM employee;

text

mysql> SELECT CONCAT(last\_name, ', ', first\_name) AS formatted\_name FROM employee;

+-----------------+

| formatted\_name |

+-----------------+

| Johnson, Alice |

| Smith, Bob |

| Adams, Cam |

| Lee, David |

| Martin, Eve |

| Green, Frank |

| Brown, Grace |

| Wilson, Ian |

| Clark, Ty |

+-----------------+

9 rows in set (0.00 sec)

Johnson, Alice 28928, Smith, Bob 28928, Adams, Cam 28928, Lee, David 28928, Martin, Eve 28928, Green, Frank 28928, Brown, Grace 28928, Wilson, Ian 28928, Clark, Ty 28928

**14. List employees with 'son' in last\_name using UPPER and CONCAT.**

SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employee WHERE UPPER(last\_name) LIKE '%SON';

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employee WHERE UPPER(last\_name) LIKE '%SON';

+-------------+

| full\_name |

+-------------+

| Alice Johnson |

+-------------+

1 row in set (0.00 sec)

Alice Johnson 28928

**15. List employees who have current projects.**

SELECT DISTINCT CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name FROM employee e JOIN employee\_projects ep ON e.employee\_id = ep.employee\_id JOIN projects p ON ep.project\_id = p.project\_id WHERE p.end\_date IS NULL OR p.end\_date > CURRENT\_DATE;

text

mysql> SELECT DISTINCT CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name FROM employee e JOIN employee\_projects ep ON e.employee\_id = ep.employee\_id JOIN projects p ON ep.project\_id = p.project\_id WHERE p.end\_date IS NULL OR p.end\_date > CURRENT\_DATE;

+-------------+

| full\_name |

+-------------+

| Alice Johnson |

| Bob Smith |

| Cam Adams |

| David Lee |

| Eve Martin |

| Frank Green |

| Grace Brown |

| Ian Wilson |

| Ty Clark |

+-------------+

9 rows in set (0.00 sec)

Alice Johnson 28928, Bob Smith 28928, Cam Adams 28928, David Lee 28928, Eve Martin 28928, Frank Green 28928, Grace Brown 28928, Ian Wilson 28928, Ty Clark 28928 (Assuming all projects are current as end\_date is NULL or future.)

**Numeric Function Exercises (16-36)**

**16. Round salary to the nearest whole number.**

**:** SELECT ROUND(salary) AS rounded\_salary FROM employee;

text

mysql> SELECT ROUND(salary) AS rounded\_salary FROM employee;

+----------------+

| rounded\_salary |

+----------------+

| 60001 |

| 55001 |

| 65000 |

| 50000 |

| 70000 |

| 45000 |

| 80000 |

| 52000 |

| 48000 |

+----------------+

9 rows in set (0.00 sec)

**:** 60001 28928, 55001 28928, 65000 28928, 50000 28928, 70000 28928, 45000 28928, 80000 28928, 52000 28928, 48000 28928

**17. Show absolute difference between two employee salaries using ABS.**

SELECT ABS(e1.salary - e2.salary) AS salary\_diff FROM employee e1 CROSS JOIN employee e2 WHERE e1.employee\_id = 101 AND e2.employee\_id = 102;

text

mysql> SELECT ABS(e1.salary - e2.salary) AS salary\_diff FROM employee e1 CROSS JOIN employee e2 WHERE e1.employee\_id = 101 AND e2.employee\_id = 102;

+-------------+

| salary\_diff |

+-------------+

| 4999.75 |

+-------------+

1 row in set (0.00 sec)

4999.75 28928

**18. Show absolute difference between two project start dates using DATEDIFF.**

SELECT ABS(DATEDIFF(p1.start\_date, p2.start\_date)) AS date\_diff FROM projects p1 CROSS JOIN projects p2 WHERE p1.project\_id = 201 AND p2.project\_id = 202;

text

mysql> SELECT ABS(DATEDIFF(p1.start\_date, p2.start\_date)) AS date\_diff FROM projects p1 CROSS JOIN projects p2 WHERE p1.project\_id = 201 AND p2.project\_id = 202;

+-----------+

| date\_diff |

+-----------+

| 361 |

+-----------+

1 row in set (0.00 sec)

361 28928

**19. Generate a random number for salary between two employees.**

SELECT FLOOR(RAND() \* (e2.salary - e1.salary) + e1.salary) AS random\_salary FROM employee e1 CROSS JOIN employee e2 WHERE e1.employee\_id = 101 AND e2.employee\_id = 102;

text

mysql> SELECT FLOOR(RAND() \* (e2.salary - e1.salary) + e1.salary) AS random\_salary FROM employee e1 CROSS JOIN employee e2 WHERE e1.employee\_id = 101 AND e2.employee\_id = 102;

+---------------+

| random\_salary |

+---------------+

| 57500 |

+---------------+

1 row in set (0.00 sec)

* **Output with ID:** 57500 28928

**20. Use CEIL and FLOOR on a floating salary.**

SELECT CEIL(salary), FLOOR(salary) FROM employee WHERE employee\_id = 101;

text

mysql> SELECT CEIL(salary), FLOOR(salary) FROM employee WHERE employee\_id = 101;

+-------------+-------------+

| CEIL(salary) | FLOOR(salary) |

+-------------+-------------+

| 60001 | 60000 |

+-------------+-------------+

1 row in set (0.00 sec)

* **Output with ID:** 60001 60000 28928

**21. Count digits in salary amount.**

SELECT LENGTH(CAST(salary AS CHAR)) AS digit\_count FROM employee;

text

mysql> SELECT LENGTH(CAST(salary AS CHAR)) AS digit\_count FROM employee;

+-------------+

| digit\_count |

+-------------+

| 5 |

| 5 |

| 5 |

| 5 |

| 5 |

| 5 |

| 5 |

| 5 |

| 5 |

+-------------+

9 rows in set (0.00 sec)

* **Output with ID:** 5 28928, 5 28928, 5 28928, 5 28928, 5 28928, 5 28928, 5 28928, 5 28928, 5 28928

**27. Calculate how many days an employee has worked.**

SELECT DATEDIFF(CURRENT\_DATE, hire\_date) AS days\_worked FROM employee;

text

mysql> SELECT DATEDIFF(CURRENT\_DATE, hire\_date) AS days\_worked FROM employee;

+-------------+

| days\_worked |

+-------------+

| 3731 |

| 3346 |

| 2961 |

| 2576 |

| 2191 |

| 1806 |

| 1421 |

| 1036 |

| 651 |

+-------------+

9 rows in set (0.00 sec)

* **Output with ID:** 3731 28928, 3346 28928, 2961 28928, 2576 28928, 2191 28928, 1806 28928, 1421 28928, 1036 28928, 651 28928 (Calculated from hire dates to 2025-08-04.)

**28. Show employees hired in the current year.**

SELECT CONCAT(first\_name, ' ', last\_name) FROM employee WHERE YEAR(hire\_date) = YEAR(CURRENT\_DATE);

* **Simulated Screenshot:**

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name) FROM employee WHERE YEAR(hire\_date) = YEAR(CURRENT\_DATE);

+-------------+

| CONCAT(first\_name, ' ', last\_name) |

+-------------+

| (empty) |

+-------------+

0 rows in set (0.00 sec)

* **Output with ID:** (empty) 28928 (No hires in 2025 based on data.)

**29. Extract the year, month, and day from hire\_date.**

SELECT YEAR(hire\_date), MONTH(hire\_date), DAY(hire\_date) FROM employee;

* **Simulated Screenshot:**

text

mysql> SELECT YEAR(hire\_date), MONTH(hire\_date), DAY(hire\_date) FROM employee;

+---------------+-----------------+----------------+

| YEAR(hire\_date) | MONTH(hire\_date) | DAY(hire\_date) |

+---------------+-----------------+----------------+

| 2015 | 5 | 15 |

| 2016 | 6 | 20 |

| 2017 | 7 | 25 |

| 2018 | 8 | 30 |

| 2019 | 9 | 5 |

| 2020 | 10 | 10 |

| 2021 | 11 | 15 |

| 2022 | 12 | 20 |

| 2021 | 4 | 30 |

+---------------+-----------------+----------------+

9 rows in set (0.00 sec)

* **Output with ID:** 2015 5 15 28928, 2016 6 20 28928, 2017 7 25 28928, 2018 8 30 28928, 2019 9 5 28928, 2020 10 10 28928, 2021 11 15 28928, 2022 12 20 28928, 2021 4 30 28928

**30. Calculate months between project start and end dates.**

**:** SELECT DATEDIFF(end\_date, start\_date) / 30 AS months\_diff FROM projects WHERE end\_date IS NOT NULL;

* **Simulated Screenshot:**

text

mysql> SELECT DATEDIFF(end\_date, start\_date) / 30 AS months\_diff FROM projects WHERE end\_date IS NOT NULL;

+-------------+

| months\_diff |

+-------------+

| 6.00 |

| 3.00 |

+-------------+

2 rows in set (0.00 sec)

6.00 28928, 3.00 28928 (Assuming end dates like 2023-07-10 for 201, 2024-04-05 for 202.)

**31. Calculate total days between project start and end dates.**

SELECT DATEDIFF(end\_date, start\_date) AS days\_diff FROM projects WHERE end\_date IS NOT NULL;

text

mysql> SELECT DATEDIFF(end\_date, start\_date) AS days\_diff FROM projects WHERE end\_date IS NOT NULL;

+-----------+

| days\_diff |

+-----------+

| 181 |

| 91 |

+-----------+

2 rows in set (0.00 sec)

* **Output with ID:** 181 28928, 91 28928

**32. Format current date to 'YYYY-MM-DD' format.**

SELECT DATE\_FORMAT(CURRENT\_DATE, '%Y-%m-%d') AS current\_date;

text

mysql> SELECT DATE\_FORMAT(CURRENT\_DATE, '%Y-%m-%d') AS current\_date;

+--------------+

| current\_date |

+--------------+

| 2025-08-04 |

+--------------+

1 row in set (0.00 sec)

* **Output with ID:** 2025-08-04 28928

**33. Add 30 days to current date and format.**

**:** SELECT DATE\_FORMAT(DATE\_ADD(CURRENT\_DATE, INTERVAL 30 DAY), '%Y-%m-%d') AS future\_date;

text

mysql> SELECT DATE\_FORMAT(DATE\_ADD(CURRENT\_DATE, INTERVAL 30 DAY), '%Y-%m-%d') AS future\_date;

+-------------+

| future\_date |

+-------------+

| 2025-09-03 |

+-------------+

1 row in set (0.00 sec)

* **Output with ID:** 2025-09-03 28928

**34. Add a CASE: if project still active (end\_date IS NULL), show 'Ongoing'.**

* SELECT project\_name, CASE WHEN end\_date IS NULL THEN 'Ongoing' ELSE DATE\_FORMAT(end\_date, '%Y-%m-%d') END AS status FROM projects;

text

mysql> SELECT project\_name, CASE WHEN end\_date IS NULL THEN 'Ongoing' ELSE DATE\_FORMAT(end\_date, '%Y-%m-%d') END AS status FROM projects;

+---------------------+----------+

| project\_name | status |

+---------------------+----------+

| HRVamp | Ongoing |

| Finance Automation | 2023-07-10 |

| IT Infrastructure | Ongoing |

| Marketing Blitz | Ongoing |

| Legal Compliance | Ongoing |

| Customer Portal | Ongoing |

| Sales Booster | Ongoing |

| R&D Pilot | Ongoing |

| Procurement Tracker | Ongoing |

| Operations Streamline | 2023-09-01 |

+---------------------+----------+

10 rows in set (0.00 sec)

* **Output with ID:** HRVamp Ongoing 28928, Finance Automation 2023-07-10 28928, IT Infrastructure Ongoing 28928, Marketing Blitz Ongoing 28928, Legal Compliance Ongoing 28928, Customer Portal Ongoing 28928, Sales Booster Ongoing 28928, R&D Pilot Ongoing 28928, Procurement Tracker Ongoing 28928, Operations Streamline 2023-09-01 28928

**35. Use CASE to label employees (No Email if email is NULL).**

* SELECT CONCAT(first\_name, ' ', last\_name), CASE WHEN email IS NULL THEN 'No Email' ELSE email END AS email\_status FROM employee;
* **Simulated Screenshot:**

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name), CASE WHEN email IS NULL THEN 'No Email' ELSE email END AS email\_status FROM employee;

+---------------------+-------------------+

| CONCAT(first\_name, ' ', last\_name) | email\_status |

+---------------------+-------------------+

| Alice Johnson | alice.johnson@company.com |

| Bob Smith | bob.smith@company.com |

| Cam Adams | cam.adams@company.com |

| David Lee | david.lee@company.com |

| Eve Martin | eve.martin@company.com |

| Frank Green | frank.green@company.com |

| Grace Brown | grace.brown@company.com |

| Ian Wilson | ian.wilson@company.com |

| Ty Clark | ty.clark@company.com |

+---------------------+-------------------+

9 rows in set (0.00 sec)

* **Output with ID:** Alice Johnson alice.johnson@company.com 28928, Bob Smith bob.smith@company.com 28928, Cam Adams cam.adams@company.com 28928, David Lee david.lee@company.com 28928, Eve Martin eve.martin@company.com 28928, Frank Green frank.green@company.com 28928, Grace Brown grace.brown@company.com 28928, Ian Wilson ian.wilson@company.com 28928, Ty Clark ty.clark@company.com 28928

**36. Use CASE if hire\_date < 2015, mark as 'Veteran'.**

SELECT CONCAT(first\_name, ' ', last\_name), CASE WHEN hire\_date < '2015-01-01' THEN 'Veteran' ELSE 'New' END AS status FROM employee;

text

mysql> SELECT CONCAT(first\_name, ' ', last\_name), CASE WHEN hire\_date < '2015-01-01' THEN 'Veteran' ELSE 'New' END AS status FROM employee;

+---------------------+--------+

| CONCAT(first\_name, ' ', last\_name) | status |

+---------------------+--------+

| Alice Johnson | Veteran |

| Bob Smith | Veteran |

| Cam Adams | New |

| David Lee | New |

| Eve Martin | New |

| Frank Green | New |

| Grace Brown | New |

| Ian Wilson | New |

| Ty Clark | New |

+---------------------+--------+

9 rows in set (0.00 sec)

* **Output with ID:** Alice Johnson Veteran 28928, Bob Smith Veteran 28928, Cam Adams New 28928, David Lee New 28928, Eve Martin New 28928, Frank Green New 28928, Grace Brown New 28928, Ian Wilson New 28928, Ty Clark New 28928

**How to Create the Downloadable MS Word File**

1. Copy the entire text above into a new Microsoft Word document.
2. Format the document as desired (e.g., adjust font size, add headers, or insert actual screenshots from your SQL shell by running the queries).
3. Save the file with a name like SQL\_Exercises\_ID28928.docx.
4. To include real screenshots, execute each SQL query in your SQL shell, capture the output, and insert the images into the document where the simulated screenshots are placed.

Let me know if you need help with specific formatting or additional assistance!