

-- 1. Write a SQL query to remove the details of an employee whose first name ends in 'even'

```
DELETE FROM EMPLOYEES  
WHERE FIRST_NAME LIKE '%even';
```

```
SELECT * FROM EMPLOYEES  
WHERE FIRST_NAME LIKE '%even';
```

-- 2. Write a query in SQL to show the three minimum values of the salary from the table.

```
SELECT SALARY FROM EMPLOYEES  
ORDER BY SALARY ASC  
LIMIT 3;
```

-- 3. Write a SQL query to copy the details of this table into a new table with table name as Employee table and to delete the records in employees table.

```
CREATE TABLE EMPLOYEE LIKE EMPLOYEES;  
SELECT * FROM EMPLOYEE;  
INSERT INTO EMPLOYEE SELECT * FROM EMPLOYEES;  
DELETE FROM EMPLOYEES;  
DROP TABLE EMPLOYEE;  
--(OR)--
```

```
CREATE TABLE EMPLOYEE AS (SELECT * FROM EMPLOYEES );
```

```
--(OR)--
```

```
CREATE TABLE IF NOT EXISTS EMPLOYEE AS (SELECT * FROM EMPLOYEES  
limit 12);
```

-- 4. Write a SQL query to remove the column Age from the table.

```
ALTER TABLE EMPLOYEE  
DROP COLUMN first_name;  
SELECT * FROM EMPLOYEE;
```

-- 5. Obtain the list of employees (their full name, email, hire_year) where they have joined the firm before 2000.

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS FULL_NAME, EMAIL,  
YEAR(HIRE_DATE) AS HIRE_YEAR FROM EMPLOYEES  
WHERE HIRE_YEAR < 2000;
```

-- 6. Fetch the employee_id and job_id of those employees whose start year lies in the range of 1990 and 1999.

```
SELECT EMPLOYEE_ID, JOB_ID FROM EMPLOYEES  
WHERE YEAR(HIRE_DATE) > 1990 AND YEAR(HIRE_DATE) < 1999;
```

```
SELECT * FROM EMPLOYEES;
```

-- 7. Find the first occurrence of the letter 'A' in each employees Email ID Return the employee_id, email id and the letter position.

```
SELECT EMPLOYEE_ID, EMAIL, CHARINDEX('A', EMAIL) AS LETTER_POSITION  
FROM EMPLOYEES;
```

-- 8. Fetch the list of employees(Employee_id, full name, email) whose full name holds characters less than 12.

```
SELECT EMPLOYEE_ID, CONCAT(FIRST_NAME, ' ', LAST_NAME) AS FULL_NAME,  
EMAIL FROM EMPLOYEES  
WHERE LENGTH(FULL_NAME) < 12;
```

-- 9. Create a unique string by hyphenating the first name, last name , and email of the employees to obtain a new field named UNQ_ID Return the employee_id, and their corresponding UNQ_ID.

```
ALTER TABLE EMPLOYEES ADD UNQ_ID VARCHAR(100);  
UPDATE EMPLOYEES SET UNQ_ID = CONCAT_WS('-', FIRST_NAME,  
LAST_NAME, EMAIL);  
SELECT EMPLOYEE_ID, UNQ_ID FROM EMPLOYEES;
```

-- 10. Write a SQL query to update the size of email column to 30.

```
ALTER TABLE EMPLOYEES MODIFY EMAIL VARCHAR(30);
```

-- 11. Write a SQL query to change the location of Diana to London.

```
UPDATE TABLE DEPARTMENTS  
SET LOCATION_ID = 2400  
WHERE DEPARTMENT_ID = (  
    SELECT DEPARTMENT_ID FROM EMPLOYEES  
    WHERE FIRST_NAME = 'Diana'  
);
```

```
SELECT * FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 60;
```

-- 12. Fetch all employees with their first name , email , phone (without extension part) and extension (just the extension) Info : this mean you need to separate phone into 2 parts eg: 123.123.1234.12345 => 123.123.1234 and 12345 . first half in phone column and second half in extension column.

```
SELECT  
FIRST_NAME,  
EMAIL,  
PHONE_NUMBER,
```

```

CASE
  WHEN LENGTH(PHONE_NUMBER) = 12 THEN SUBSTR(PHONE_NUMBER,1,7)
  WHEN LENGTH(PHONE_NUMBER) = 18 THEN SUBSTR(PHONE_NUMBER,1,11)
  ELSE SUBSTR(PHONE_NUMBER,1,9)
END AS PHONE,
SPLIT_PART(PHONE_NUMBER, '-',1) AS EXTENSION
FROM EMPLOYEES;

```

-- 13. Write a SQL query to find the employee with second and third maximum salary with and without using top/limit keyword.

```

SELECT TOP 2 SALARY FROM (
  SELECT TOP 3 SALARY FROM EMPLOYEES
  ORDER BY SALARY DESC
) ORDER BY SALARY ASC;

```

-- 14. Fetch all details of top 3 highly paid employees who are in department Shipping and IT.

```

SELECT * FROM EMPLOYEES
WHERE DEPARTMENT_ID = 50 OR DEPARTMENT_ID = 60
ORDER BY SALARY DESC
LIMIT 3;

```

-- 15. Display employee id and the positions(jobs) held by that employee (including the current position).

```

SELECT EMPLOYEE_ID, JOB_ID FROM EMPLOYEES UNION SELECT
EMPLOYEE_ID, JOB_ID FROM JOB_HISTORY ORDER BY EMPLOYEE_ID;
SELECT * FROM JOB_HISTORY;

```

-- 16. Display Employee first name and date joined as WeekDay, Month Day, Year Eg : Emp ID | Date Joined -> 1 | Monday, June 21st, 1999

```

SELECT CONCAT(DAYNAME(HIRE_DATE),',', MONTHNAME(HIRE_DATE),',',
DAY(HIRE_DATE),'th',',',YEAR(HIRE_DATE))) AS FORMATTED_DATE
FROM EMPLOYEES;
SELECT DATE_FORMAT(HIRE_DATE, '%M %D %Y') AS formatted_date
FROM EMPLOYEES;
-- SELECT 101 AS EMPLOYEE_ID
-- UNION
-- SELECT
-- EMPLOYEE_ID FROM JOB_HISTORY;

```

-- 17. The company holds a new job opening for Data Engineer (DT_ENGG) with a minimum salary of 12,000 and maximum salary of 30,000 .

- The job position might be removed based on market trends (so, save the changes) .
- - Later, update the maximum salary to 40,000 .
- - Save the entries as well.

```
-- - Now, revert back the changes to the initial state, where the salary was 30,000
ALTER SESSION SET AUTOCOMMIT = FALSE;
INSERT INTO JOBS VALUES('DT_ENGG','DATA ENGINEER', 12000, 30000);
COMMIT;
UPDATE JOBS SET MAX_SALARY = 40000 WHERE JOB_ID = 'DT_ENGG';
ROLLBACK;
SELECT * FROM JOBS;
-- DELETE FROM JOBS WHERE JOB_ID = 'DT_ENGG';
```

-- 18. Find the average salary of all the employees who got hired after 8th January 1996 but before 1st January 2000 and round the result to 3 decimals

```
SELECT ROUND(AVG(SALARY), 3) AS AVERAGE_SALARY
FROM EMPLOYEES
WHERE HIRE_DATE > '1996-01-08' AND HIRE_DATE < '2000-01-01';
```

-- 19. Display Australia, Asia, Antarctica, Europe along with the regions in the region table
(Note: Do not insert data into the table)

-- 19A Display all the regions

```
SELECT REGION_NAME FROM REGIONS
UNION SELECT 'Australia'
UNION SELECT 'Asia'
UNION SELECT 'Antarctica'
UNION SELECT 'Europe';
```

-- 19B Display all the unique regions

```
SELECT DISTINCT REGION_NAME FROM REGIONS;
--SELECT * FROM REGIONS;
```

-- 20. Write a SQL query to remove the employees table from the database

```
DROP TABLE EMPLOYEES;
SELECT * FROM EMPLOYEES;
```