

TOP SQL QUESTIONS

PART 4

Give 20% salary hike to employees who has completed 5 years of service in the organization?

```
CREATE TABLE EMPLOYEE  
(  
  ID INT,  
  NAME VARCHAR (40),  
  DEPT VARCHAR (40),  
  DATE_OF_JOINING DATE,  
  SALARY INT  
);
```

```
INSERT INTO EMPLOYEE VALUES  
(1,'A', 'SALES','2020-06-08',40000),  
(2,'B', 'ADMIN','2022-11-11' ,50000),  
(3, 'C','ANALYST','2016-05-22',70000),  
(4,'D','HR','2022-12-30' ,60000),  
(5, 'E','R&D','2020-09-20',60000),  
(6,'F','SALES','2018-04-10' ,30000);
```

```
SELECT * FROM EMPLOYEE;
```

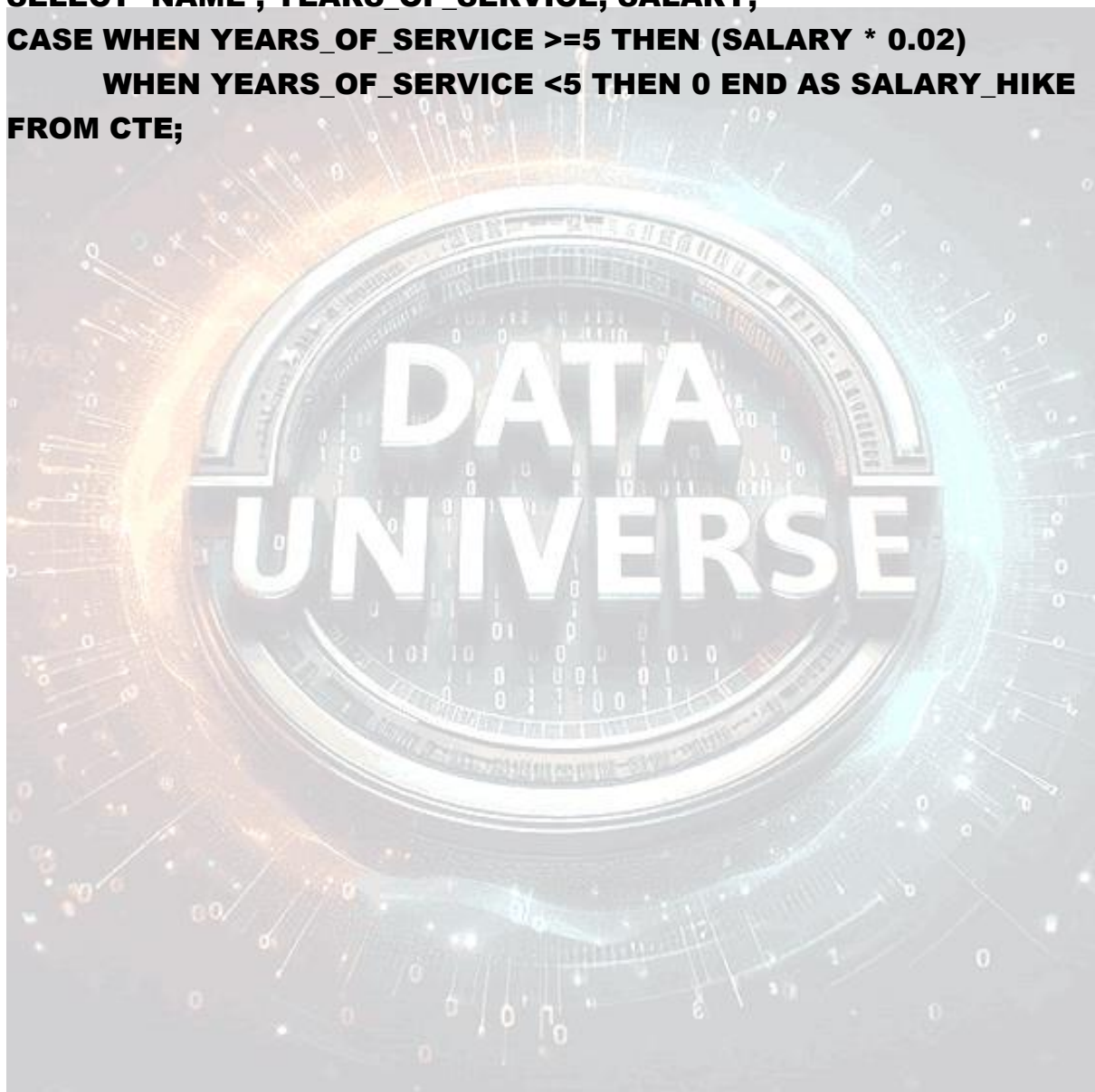
/*----TIMESTAMPDIFF use to calculate the difference between the timestamps in MySQL*/

```
SELECT *,  
TIMESTAMPDIFF (YEAR, DATE_OF_JOINING, CURRENT_DATE ()) AS  
YEARS_OF_SERVICE  
FROM EMPLOYEE;
```

WITH CTE AS

```
(  
SELECT *, TIMESTAMPDIFF (YEAR, DATE_OF_JOINING,  
CURRENT_DATE ()) AS YEARS_OF_SERVICE FROM EMPLOYEE  
)
```

```
SELECT `NAME`, YEARS_OF_SERVICE, SALARY,  
CASE WHEN YEARS_OF_SERVICE >=5 THEN (SALARY * 0.02)  
      WHEN YEARS_OF_SERVICE <5 THEN 0 END AS SALARY_HIKE  
FROM CTE;
```



Find cumulative sum or running total of revenue for each product category?

CREATE TABLE PRODUCTS

(
PRODUCT_ID INT,
PRODUCT_CATEGORY VARCHAR (40),
PRODUCT_NAME VARCHAR (40),
REVENUE INT
);

INSERT INTO PRODUCTS VALUES

(1, 'FURNITURE','CHAIR', 6000),
(2,'ELECTRONICS','FAN',80000),
(3,'BEAUTY','SHAMPOO', 20000),
(4,'FASHION','T-SHIRT',30000),
(5,'GROCERIES','FRUITS',40000),
(6,'MOBILE','SAMSUNG',300000),
(7,'BEAUTY','SOAP', 50000),
(8,'FASHION','JEANS',250000),
(9,'GROCERIES','DAIRY',40000),
(10,'MOBILE','APPLE',500000);

SELECT * FROM PRODUCTS;

SELECT *, SUM (REVENUE) OVER (PARTITION BY
PRODUCT_CATEGORY ORDER BY PRODUCT_ID)
AS CUMULATIVE_SUM FROM PRODUCTS;

FIND RETIREMENT DATE FOR EACH EMPLOYEE AND ADD THIS COLUMN IN A TABLE?

```
CREATE TABLE EMPLOYEE (  
ID INT,  
NAME VARCHAR (40),  
DEPT VARCHAR (40),  
BIRTHDATE DATE,  
SALARY INT);
```

```
INSERT INTO EMPLOYEE VALUES  
(1,'A', 'SALES','1970-06-08',40000),  
(2,'B', 'ADMIN','1977-11-11' ,50000),  
(3, 'C','ANALYST','1972-05-22',70000),  
(4,'D','HR','1980-12-30' ,60000),  
(5, 'E','R&D','2000-09-20',60000),  
(6,'F','SALES','1990-04-10' ,30000);
```

```
SELECT * FROM EMPLOYEE;
```

```
/* -----CONSIDER RETIREMENT AGE IS 60 YEARS-----*/
```

```
/*--DATE_ADD () function adds a time/date interval to a date and then  
returns the date
```

```
-----DATE_ADD (date, INTERVAL value add unit) -----*/
```

```
SELECT *, DATE_ADD (BIRTHDATE, INTERVAL 60 YEAR) AS  
RETIREMENT_DATE FROM EMPLOYEE;
```

```
ALTER TABLE EMPLOYEE  
ADD COLUMN RETIREMENT_DATE DATE;
```

```
UPDATE EMPLOYEE  
SET RETIREMENT_DATE= DATE_ADD (BIRTHDATE, INTERVAL 60  
YEAR);  
SELECT * FROM EMPLOYEE;
```


PATTERN MATCHING USING LIKE OPERATOR IN SQL?

```
CREATE TABLE COUNTRIES  
(  
  COUNTRY_ID INT,  
  COUNTRY_NAME VARCHAR (40)  
);
```

```
INSERT INTO COUNTRIES VALUES  
(1, 'UNITED STATES'),  
(2, 'INDIA'),  
(3, 'RUSSIA'),  
(4, 'CHINA'),  
(5, 'GERMANY'),  
(6, 'UNITED KINGDOM'),  
(7, 'FRANCE');
```

```
SELECT * FROM COUNTRIES;
```

```
/*-----FIND COUNTRIES NAME STARTING WITH 'U' -----*/
```

```
SELECT COUNTRY_NAME FROM COUNTRIES WHERE  
COUNTRY_NAME LIKE 'U%';
```

```
/*---FIND COUNTRIES NAME ENDING WITH 'A'-----*/
```

```
SELECT COUNTRY_NAME FROM COUNTRIES WHERE  
COUNTRY_NAME LIKE '%A';
```

```
/*---FIND COUNTRIES NAME WITH 'S' ANYWHERE -----*/
```

```
SELECT COUNTRY_NAME FROM COUNTRIES WHERE  
COUNTRY_NAME LIKE '%S%';
```

```
/*---FIND COUNTRIES NAME WITH 'IN' ANYWHERE -----*/
```

**SELECT COUNTRY_NAME FROM COUNTRIES WHERE
COUNTRY_NAME LIKE '%IN%';**

/*---FIND COUNTRIES NAME WITH 'N' IN 2ND PLACE-----*/

**SELECT COUNTRY_NAME FROM COUNTRIES WHERE
COUNTRY_NAME LIKE '_N%';**

**/*---FIND COUNTRIES NAME START WITH 'U' AND END WITH
'M'-----*/**

**SELECT COUNTRY_NAME FROM COUNTRIES WHERE
COUNTRY_NAME LIKE 'U%M';**

**/*---FIND COUNTRIES NAME STARTS WITH 'F' & HAS ATLEAST
4 CHARACTERS IN LENGTH -----*/**

**SELECT COUNTRY_NAME FROM COUNTRIES WHERE
COUNTRY_NAME LIKE 'F____%';**

**/*---FIND COUNTRIES NAME THAT DO NOT STARTS WITH 'U'-----
-----*/**

**SELECT COUNTRY_NAME FROM COUNTRIES WHERE
COUNTRY_NAME NOT LIKE 'U%';**

➤ **EXPLAIN PRIMARY KEY IN SQL WITH PRACTICAL?**

CREATE TABLE STUDENT_RECORDS

**(
ID INT PRIMARY KEY,
`NAME` VARCHAR (40),
BRANCH VARCHAR (40),
EMAIL_ID VARCHAR (40)
);**

DESCRIBE STUDENT_RECORDS;

**INSERT INTO STUDENT_RECORDS VALUES (1,'A',
'COMPUTER', 'a@gmail.com'),
(2,'B', 'ELECTRONICS', 'b@gmail.com'),
(3,'C', 'CIVIL', 'c@gmail.com'),
(4,'A', 'ELECTRICAL', 'aa@gmail.com');**

SELECT * FROM STUDENT_RECORDS;

**INSERT INTO STUDENT_RECORDS VALUES (1,'D', 'CIVIL',
'd@gmail.com');**

**INSERT INTO STUDENT_RECORDS VALUES (5,'D', 'CIVIL',
'd@gmail.com');**

**INSERT INTO STUDENT_RECORDS (`NAME`, BRANCH,
EMAIL_ID) VALUES ('E', 'CIVIL', 'e@gmail.com');**

**INSERT INTO STUDENT_RECORDS (ID, `NAME`, BRANCH,
EMAIL_ID) VALUES (6,'E', 'CIVIL', 'e@gmail.com');**

ALTER TABLE STUDENT_RECORDS

**ADD CONSTRAINT EMAIL_CONSTRAINT PRIMARY
KEY(EMAIL_ID);**

-----PRIMARY KEY CREATED USING MULTIPLE COLUMN-----

CREATE TABLE STUDENT_RECORDS1

**(
ID INT,
`NAME` VARCHAR (40),
BRANCH VARCHAR (40),
EMAIL_ID VARCHAR (40),
PRIMARY KEY (ID, `NAME`)
);**

**-----PRIMARY KEY AUTOMATICALLY CREATES CLUSTER
INDEX -----**

SHOW INDEX FROM STUDENT_RECORDS1;