Coding Styles (Professional):

**SQL Best Practices:**

**Coding Techniques and Best Practices**

**complying with coding style is crucial**

you will always work in a team

clean code

code that is focused and understandable, which means it must be readable, logical, and changeable

good code is not the one computer understand; it is the one human can understand

when assigning names to variables or SQL objects, always chose shorter, meaningful names, conveying specific information

names will constitute more than 80% of your code

readability

unification of coding style is a top priority

/\* … \*/ (for large comments)

# or -- (for one-line comments)

Topic 2: **SQL Aggregate Functions**

Aggregate functions are a bunch of methods that operate on a set of values. They can do calculations for us and then returns one final value.

1. COUNT function

2. MIN function

3. MAX function

4. SUM function

5. AVG function

Example:

1. CREATE TABLE empl (month INT, emp VARCHAR(15), dept VARCHAR(15), salary INT);
2. desc empl;
3. INSERT INTO empl VALUES(1, "Oliver", "HR", 9000),(1, "George", "IT", 8000),(3, "Harry", "HR", 20000),(6, "Jack", "IT", 110123),(6, "Jacob", "SALES", 3000),(12, "Noah", "SALES", 101000),(12, "Charlie", "IT", 123456);

4. SELECT \* FROM empl;

**-- Count total no. of employess**

SELECT COUNT(\*) FROM empl;

**-- Count total no. of employees in HR dept**

select count(\*) from empl where dept ='HR';

-- **Count total no. of employees in each dept**

SELECT COUNT(\*), dept FROM empl GROUP BY dept;

**-- Count total no. of employees earning 9000 or below in each dept**

SELECT COUNT(\*), dept FROM empl WHERE salary <= 9000 GROUP BY dept;

-- **Count distinct joining months in the empl table**

SELECT COUNT(DISTINCT month) FROM empl;

**MIN**

**-- Find the employee with the lowest salary**

**SELECT MIN(salary) FROM empl;**

**-- Find the lowest salaries in each dept**

**SELECT dept, MIN(salary) FROM empl GROUP BY dept;**

**-- Find month-wise minimum salaries**

**SELECT month, MIN(salary) FROM empl GROUP BY month;**

**-- Find full employee detail having the lowest salary**

**SELECT \* FROM empl**

**WHERE salary = (SELECT MIN(salary) FROM empl);**

**MAX**

**-- Find the employee with the highest salary**

**SELECT MAX(salary) FROM empl;**

**-- Find the highest salaries in each dept**

**SELECT dept, MAX(salary) FROM empl GROUP BY dept;**

**-- Find month-wise maximum salaries**

**SELECT month, MAX(salary) FROM empl GROUP BY month;**

**-- Find full employee detail having the highest salary**

**SELECT \* FROM empl**

**WHERE salary = (SELECT MAX(salary) FROM empl);**

**SUM**

**-- Find the sum all employee salaries**

**SELECT SUM(salary) FROM empl;**

**-- Find the sum of salaries in each dept**

**SELECT dept, SUM(salary) FROM empl GROUP BY dept;**

**-- Find month-wise sum of salaries**

**SELECT month, SUM(salary) FROM empl GROUP BY month;**

**AVG**

**-- Find the AVG of salaries in each dept**

**SELECT dept, AVG(salary) FROM empl GROUP BY dept;**

**-- Find month-wise AVG of salaries**

**SELECT month, AVG(salary) FROM empl GROUP BY month;**

**Topic:3**

**MySQL OPTIMIZE TABLE Statement**



**MySQL OPTIMIZE TABLE statement which defrags tables and recovers unused space. We’ll describe the complete usage of this method with the help of simple examples.**

**If your database is receiving a lot of deletes and updates calls, then it might lead to fragmentation in your MySQL data files.**

**Therefore, a lot of unused space would go in vain, and also put a high impact on the performance.**

**So, experts recommend that you must defrag your MySQL tables regularly.**

## MySQL OPTIMIZE TABLE

Before you do optimization, first confirm whether your MySQL database is suffering from fragmentation or not. To know it, run the below command.

**Check Tables for Optimization**

**You need to analyze which table is consuming more space in your database. Hence, connect to the MySQL DB instance, and run the below query.**

**It should fetch the tables which are accounting for the unused space.**

**-- List all tables causing unused space**

**SELECT TABLE\_NAME,**

**ROUND(DATA\_LENGTH/1024/1024) AS USED\_SPACE\_MB,**

**ROUND(DATA\_FREE/1024/1024) AS UNUSED\_SPACE\_MB**

**FROM INFORMATION\_SCHEMA.TABLES**

**WHERE ROUND(DATA\_FREE/1024/1024) > 1000**

**ORDER BY UNUSED\_SPACE\_MB;**

Graphical user interface, text

Description automatically generated

**We can interpret the following facts from the output:**

First, the SELECT command is listing all tables that are causing more than 1000 MB of free space.

The columns USED\_SPACE\_MB and UNUSED\_SPACE\_MB are showing data in MB.

The results indicate that all three tables are a candidate for optimization as they are causing high fragmentation.

MySQL OPTIMIZE TABLE Command:

OPTIMIZE TABLE EMPLOYEES;

OPTIMIZE TABLE EMPLOYEES, EmployeeUSA1, EmployeeUK1;

While running optimization on a table, MySQL does the following tasks:

Creates a temp table,

Deletes the original one after optimizing it, and

Rename the temp table to the original name in the end.

**Post-MySQL Optimization:**

After finishing up with optimization, you can issue the below command. It will fetch the size of the total as well as the unused-space the three tables are claiming.

**-- Query tables we'd optimized**

**SELECT TABLE\_NAME,**

**ROUND(DATA\_LENGTH/1024/1024) AS USED\_SPACE\_MB,**

**ROUND(DATA\_FREE/1024/1024) AS UNUSED\_SPACE\_MB**

**FROM INFORMATION\_SCHEMA.TABLES**

**WHERE TABLE\_NAME in**

**('EMPLOYEES', 'SALESINFO', 'FINANCES');**

**After running the above SQL query, you shall see this type of result:**

Text

Description automatically generated

You can easily deduce from the outcome that MySQL OPTIMIZE TABLE command has significantly reduced the size. And, unused space is no more.

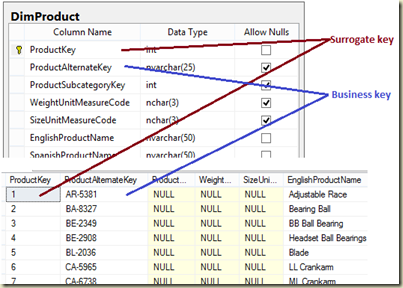
Also, the table sizes have come down too. And it helped us fix a lot of fragmentation at the filesystem level.

A picture containing shape

Description automatically generated

Text

Description automatically generated

****

A picture containing text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A picture containing text

Description automatically generated

Text

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

Text

Description automatically generated

A picture containing logo

Description automatically generated