Start-Tech Academy

Displays the execution plan for a query statement without running the query.

EXPLAIN (Syntax)

EXPLAIN [VERBOSE] *query;*

VERBOSE
Displays the full query plan instead of just a summary.
query
Query statement to explain.

Best Practices

SOFT DELETE
vs
HARD DELETE

SOFT DELETE

Soft deletion means you don't actually delete the record instead you are marking the record as deleted

HARD DELETE

Hard deletion means data is physically deleted from the database table.

Start-Tech
ACADEMY

Best Practices

UPDATE vs CASE

UPDATE

Update customer set customer_name = (trim(upper(customer_name)) where (trim(upper(customer_name)) <> customer_name

Every updated row is actually a soft delete and an insert. So updating every row will increase the storage size of the table

CASE STATEMENT

Instead you can use the case statements while creating such tables

VACCUM

SYNTAX

`VACUUM [ ` ***table*** ` ]`

USE

➢ Reclaims disk space occupied by rows that were marked for deletion by previous UPDATE and DELETE operations.

➢ Compacts the table to free up the consumed space

➢ Use it on tables which you are updating and deleting on a regular basis

Best Practices

**TRUNCATE
VS
DELETE**

➢ The TRUNCATE statement is typically far more efficient than using the DELETE statement with no WHERE clause to empty the table

➢ TRUNCATE requires fewer resources and less logging overhead

➢ Instead of creating table each time try to use truncate as it will keep the table structure and properties intact

➢ Truncate frees up space and impossible to rollback

Best Practices

STRING FUNCTIONS

Pattern Matching

➢ Whenever possible use LIKE statements in place of REGEX expressions

➢ Do not use 'Similar To' statements, instead use Like and Regex

➢ Avoid unnecessary string operations such as replace, upper, lower etc

String Operations

➢ Use trim instead of replace whenever possible

➢ Avoid unnecessary String columns. For eg. Use date formats instead of string for dates

Best Practices

JOINS

Syntax

SELECT   a.order_line ,   a.product_id,   b.customer_name,   b.age

FROM sales_2015 AS a LEFT JOIN customer_20_60 AS b

ON a.customer_id = b.customer_id

ORDER BY customer_id;

Best Practices

➢ Use subqueries to select only the required fields from the tables

➢ Avoid one to many joins by mentioning Group by clause on the matching fields

Start-Tech
ACADEMY

Best Practices

A *schema* is a collection of database objects associated with one particular database.
You may have one or multiple schemas in a database.

SCHEMAS

1. To allow many users to use one database without interfering with each other.

2. To organize database objects into logical groups to make them more manageable.

3. Third-party applications can be put into separate schemas so they do not collide with the names of other objects.

Start-Tech
ACADEMY

Best Practices

A *schema* is a collection of database objects associated with one particular database.
You may have one or multiple schemas in a database.

**SCHEMAS
(Syntax)**

CREATE SCHEMA testschema;