SQL Best Practices

Chandra Sekhar
Agenda

- Application failing with "String too large error"
- My application is not running fast enough
- Push your application logic to User Defined Functions
- Use Scalar Subquery to simplify your query
- Which Isolation level is good for your application?
PROGRESS SQL-92 DATABASE PERMISSIONS & SECURITY

- Granting permissions on tables to regular users:
  000078878, How to GRANT permissions for SQL92 to a user?
  000010288, How to grant SELECT permissions to a user?
  000022114, How to generate SQL script to GRANT all privileges on all the database tables using 4GL?
  000001611, Progress SQL-92: Grant privileges Without Creating Users?
  000039715, How to generate SQL script to GRANT Read and Write privileges on all the database tables using 4GL?

- Revoking permissions on tables to regular users:
  000067931, Basic SQL 92 REVOKE syntax.
  000022121, How to revoke a SQL92 user right to update the database?
  000014112, SQL: How to revoke SQL privileges of the Default DBA and hide Progress Schema tables from public?
  000078881, ABL/4GL Code to revoke rights to SCHEMA or ALL tables.
Application failing with "String too large error"
Application failing with "String too large error"

- It is problem with SQL Maximum column width.
  
  SQL: “ahh, (The Sellout) you are the best!!”
  SQL: “I can read but partially (The)!!”
  SQL: “You are too big for me!!”
  ABL: “I Can read you(The Sellout)!!”

```
varchar(3):
“Hey, can you read me  (The Sellout)?”
```
SQL width problem & classic solution

Select X from DB

SQL

ABL

DB Too Large

X Varchar(5)

Opt 2: scan and fix

dbtool

done

© 2017 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
SQL width problem & other solutions

- **Server side fixes**
  - Data Dictionary: Options -> SQL Properties -> Adjust Field Width
  - OE SQL as DBA: ALTER TABLE ALTER COL SET PRO_SQL_WIDTH ..... 

- **Client side fixes**
  - Find column which has size problem from error text
    - SELECT Name, city, state, country FROM pub.customer ..... 
    - “Column Name in table PUB.Customer has value exceeding its max length or precision.”
  - Change sql statement column reference to “substr()”
    - SELECT SUBSTR(Name, 1, 84) AS `Name`, city, state, country, ..... 

32K * 100 cols * 50 rows = 160M
SQL width problems & Modern solution

Select X from DB

Too Large

SQL

ABL

X Varchar(3)

DB

proserve –db <dbname> -S <port-number> -
SQLTruncateTooLarge <output/on/off/>

proserve –db <dbname> -S <port-number> -
SQLWidthUpdate <on/off>

My application is not running fast enough
Troubleshooting Performance problems

- Check whether **Statistics** are up to date or not?
- Get and analyze **Query plan**
- **Overloaded** server due to configuration
- Corner the SQL queries which are causing problems
  - **Enabling logging** – server log, connection log
  - Correlating .lg and server log files to match a particular user
- Check if there are any **changes** that happened **recently**
  - Changes in Query
  - Startup Parameters
Building the Query Plan

User

sql statement

schema

sql statistics

SQL Optimizer

Query plan

SQL Runtime
Importance of SQL Statistics

Consider a join between three tables A, B, and C

Which one is good???

Query Optimizer

SQL Engine

Without Statistics

All tables are Equal

A – Customer
B – Order
C – Orderline

A – 10k
B – 10k
C – 10k

A – 10k
B – 10k
C – 10k

© 2017 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
Update Statistics – verify and update

**Update Table Statistics and Index Statistics and All Column Statistics for PUB.Customer:**

- **Table:**
  - A – 10
  - B – 10k
  - C – 1000k

- **Column:**
  - _SysColStat

- **Index:**
  - _SysIdxStat
  - _Sys<datatype>Stat

---

**How do I update?**

**Join**

- **A**
- **B**
- **C**

**Join**

- **A – 10**
- **B – 10k**
- **C – 1000k**

---

**How often should I update?**
Analyze Query Plan

```
SELECT "_Description" FROM Pub."_Sql_Qplan"
WHERE "_Pnumber"=(SELECT MAX("_Pnumber")
FROM PUB."_Sql_Qplan" WHERE "_Ptype">0);
```

SQL query plan (_SQL_Qplan)

Last 10 queries!!

```
| JOIN [13][AUG_NESTED_LOOP-JOIN]
|  | PUB.Order. [12](
|  |  | INDEX SCAN OF (AmountDate,
|  |  |  | (PUB.O.Amount) between (...)
|  |  |  | (PEXPR3) = (PEXPR5)
|  |  | -- above defines ANL left side keys <relop> right side keys.
|  | PUB.Customer. [11](
|  |  | INDEX SCAN) OF (CustNum,
|  |  |  | (PUB.C.CustNum) = (null))
```
What to look in query plans

- Query plan and check for below operations
  - Sort and Aggregations

```sql
SELECT Custnum, COUNT(Ordernum) FROM Pub.Order GROUP BY Custnum
ORDER BY Custnum
```

CREATE INDEX custorder ON pub.Order(Custnum);
What to look in query plans

- Query plan and check for below operations
  - Sort and Aggregations
  - Dynamic index

```
|   |   |   INDEX SCAN OF DYN. INDEX ON SYSPROGRESS
|   |   |   [Dynamic Index with key == fld lis
|   |   |   [ Pexpr map for Dynamic Index data
|   |   |   PEXPR1, PEXPR2 ]
|   |   |   [ Fldids
|   |   |   FldId 0, FldId 1 ]
|   |   |   [Dynamic Index] TMPTBL000000f5BL00
|   |   |   (SYSPROGRESS.TMPTBL000000f4.0)
|   |   |   PROJECT [19] ( PUB.O. [2](
```
What to look in query plans

- Query plan and check for below operations
  - Sort and Aggregations
  - Dynamic index
  - Nested loop joins

```
|   PROJECT [21] (  
|   | JOIN [3][NESTED_LOOP-JOIN](
|   | PROJECT [16] (  
|   | | PUB.SI. [1](
|   | | | TABLE SCAN
|   | | | )
```
What to look in query plans

- Query plan and check for below operations
  - Sort and Aggregations
  - Dynamic index
  - Nested loop joins
  - Table scan

- Most of the scenarios above operations can cause performance problems

- Probably Indicates a missing index or missing predicate
Optimizer Hints

- Use SQL hints to see if it mitigates (workaround) the performance problem.

- **NO REORDER** – This will set given join order
  - If suspicious join order is selected by optimizer this hint can be used this to set the join order
  - Table1 INNER JOIN table 2 {NO REORDER}

- **FORCE INDEX** – Forces optimizer to use particular index
  - FORCE (INDEX (<index_name>))

- **NOEXECUTE** – Use this to generate query plan without execution
  - This hint will make query to go through optimizer phase
  - Execution will be skipped
    
    ```sql
    SELECT * FROM Pub.Customer NOEXECUTE;
    ```
Overloaded server due to configuration

- Example configuration: -Ma 10  –Mn 30
- New sql servers are not started due to given configuration

```
SQL Connections

SQL1  SQL2  ...  SQL30

-Ma maximum clients per server
-Mi  minimum clients per server
-Mn max number of servers

DB
```
Overloaded server due to configuration

- Example configuration: -Ma 10  –Mn 30
- New sql servers are not started due to given configuration
- Use –Mi to distribute connections
- Ma 10 –Mi  3 –Mn 3
  -Ma maximum clients per server
  -Mi  minimum clients per server
  -Mn max number of servers
Push your application logic to Database to improve network traffic
Application generate huge data flow over network

All orders in last 3 Months where it took more than 2 days to dispatch an order after receiving the order.

Order Table has million rows while only few orders satisfies the criteria

Push your application logic to Database using User Defined Functions.
User Defined Function

- JAVA based User Defined Function

**Create Function** `DiffDate(startDate Date, endDate Date)`
return integer
begin
    /* calculate date difference in java. */
    long dayDiff = ChronoUnite.DAYS.between(startDate, endDate);
    return dayDiff
end
Use Sub Query (SQ) and Scalar-SQ to simplify your query
Use SSQ to simplify your query.

- Scalar Sub Queries are very powerful tools.
- To list few, Scalar Subqueries can be used:
  - CASE expression
  - VALUE clause of INSERT statement
  - WHERE
  - ORDER BY

```sql
Select employee_id, name, salary from employees where salary > (select avg(salary) from employees);
```

```sql
SELECT first_name, last_name, department_id FROM employees WHERE department_id NOT IN (SELECT DEPARTMENT_ID FROM departments WHERE manager_id BETWEEN 100 AND 200);
```

Progress OpenEdge SQL Compliance to Core SQL:2011

What’s good Isolation level for your application?
DIRTY READS

Thread 1

Col 50

SELECT Col

Col = 100

Update Col

Col 100

Roll Back

Thread 2

SELECT Col

Col 100
NON REPEATABLE READS

Thread 1

Col 50

SELECT Col

Col = 100

Update Col

Commit

Thread 2

SELECT Col

Col 50

SELECT Col

Col 100
PHANTOM READS

Thread 1

N
Count(Customers)

N+1
Count(Customers)

Thread 2

INSERT Customers
What’s good Isolation level for your application?

<table>
<thead>
<tr>
<th>Isolation level</th>
<th>Dirty reads</th>
<th>Non-repeatable reads</th>
<th>Phantoms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Uncommitted</td>
<td>may occur</td>
<td>may occur</td>
<td>may occur</td>
</tr>
<tr>
<td>Read Committed</td>
<td>don't occur</td>
<td>may occur</td>
<td>may occur</td>
</tr>
<tr>
<td>Repeatable Read</td>
<td>don't occur</td>
<td>don't occur</td>
<td>may occur</td>
</tr>
<tr>
<td>Serializable</td>
<td>don't occur</td>
<td>don't occur</td>
<td>don't occur</td>
</tr>
</tbody>
</table>
Questions?