A Practical Application of Identifying and Correcting P6 Corrupt Data

Ronald M. Winter, PSP FAACE and
Marina G. Sominsky, PSP

Abstract

The integrity of the data in Primavera P6 database is important, but what does that mean in real life to a typical scheduler? Some issues relate to problems in the system, while others come imbedded in the schedule itself. Imbedded errors move with the schedule when it is copied or backed up, causing the error to propagate.

Parts 1 and 2 of the papers on corruption of P6 schedule databases relied heavily on the theory behind the process. This paper focuses on the practical manifestations of what a corrupt P6 schedule looks like and proper procedures how to repair it that any scheduler can perform. Issues such as scheduling errors are illustrated and why they matter to the average scheduler are explained.

The likelihood of Primavera P6 schedules having these corruptions is tested using an all-new, larger database. Updated benchmarks on P6 schedule database corruptions are presented in an easy to understand format. The issues pertaining to the overall system (which typically require IT intervention) are separated from those affecting individual schedules (that individual schedulers can fix).

Introduction

The integrity of the data in Oracle/Primavera EPPM or P6 Professional (P6) database is important, but what does that mean in real life to a typical scheduler? Some issues can be attributed to problems caused by the software and others - by the schedule file itself. The latter migrate with the schedule during import/export and when the file is copied and backed up.

Parts 1 and 2 of the papers on corruption of P6 schedule databases rely heavily on the theory behind the process. This paper focuses on the practical manifestations of what a corrupt P6 schedule looks like and proper procedures how to repair it that any scheduler can perform. Issues such as scheduling errors are illustrated and why they matter to the average scheduler are explained.

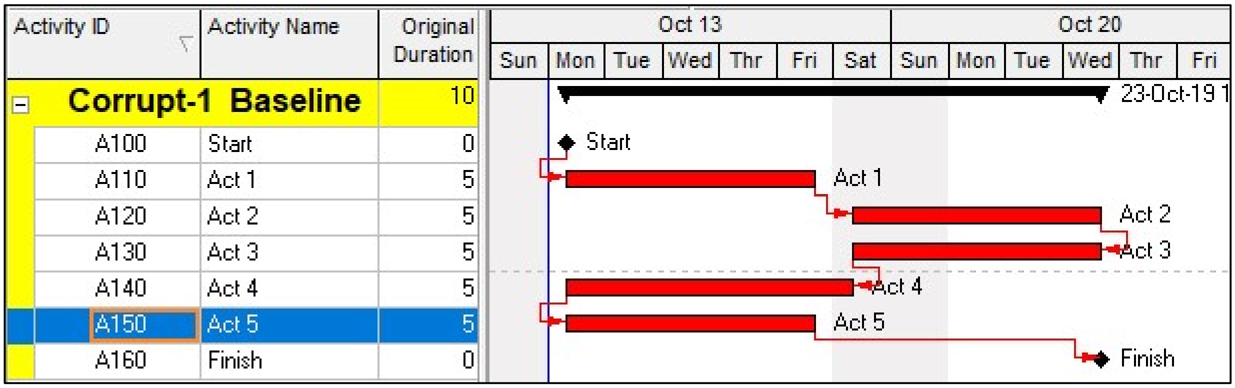
The likelihood of P6 schedules having these corruptions is investigated using an all-new, larger database. Updated benchmarks on P6 schedule database corruptions are presented in an easy to understand format. The issues pertaining to the overall system (which typically require IT intervention) are separated from those affecting individual schedules (that individual schedulers can fix).

Corrupt / Missing Relationship Information

Corrupt and missing information in P6 relationship records occurs across the industry. Earlier studies suggest that forty-five percent (45%) of all schedules have some sort of database corruption. [1] Same studies determine that a missing lag value for relationships is one of the largest categories of corruption types, wherein eighty-seven percent (87%) of examined schedules in fifteen (15) different P6 databases have missing relationship lags.

Another corruption type in the above studies - unrecognized relationship types - affects seven percent (7%) of schedules. Despite seemingly low number of databases with the latter corruption, this is not a negligible result, because the critical path computation cannot truly be made without legitimate relationship table records.

The earlier studies do not explain how Oracle Primavera P6 reacts to an existing relationship corruption, nor the effect on critical path method (CPM) computations. To investigate and illustrate this phenomenon, the authors created a simple CPM schedule using all four types of relationships and a lag. The graphical representation of the CPM network is shown in Figure 1 below.



Activity ID	Activity Name	Original Duration
Corrupt-1 Baseline		10
A100	Start	0
A110	Act 1	5
A120	Act 2	5
A130	Act 3	5
A140	Act 4	5
A150	Act 5	5
A160	Finish	0

Figure 1 – Critical Path Method (CPM) Baseline Schedule

The relationships in the baseline schedule are defined by the predecessor activity (PRED), type (TYPE), and successor activity (SUCC). Six relationships (REL) are created and listed in Table 1:

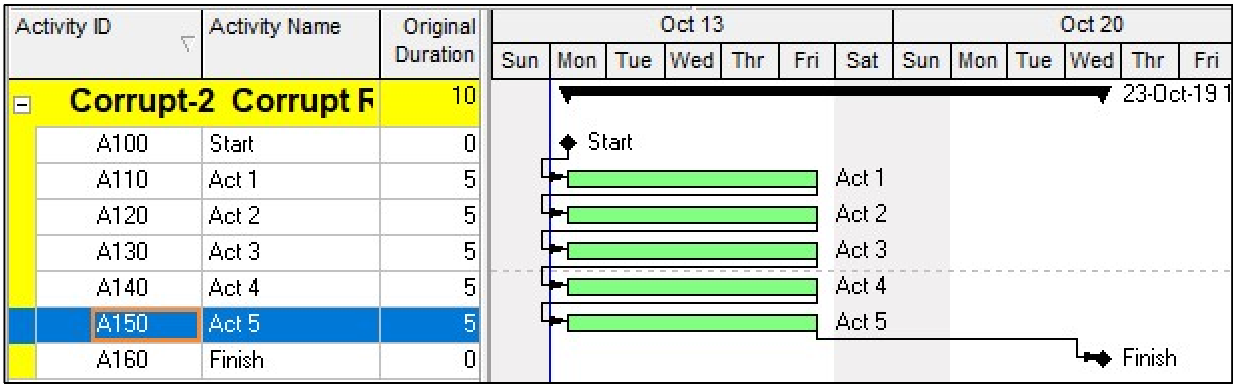
REL	PRED	TYPE\LAG	SUCC
1	A100	FS\0	A110
2	A110	FS\0	A120
3	A120	FF\0	A130
4	A130	SF\0	A140
5	A140	SS\0	A150
6	A150	FS\5	A160

Table 1 – Baseline Schedule Relationship Definitions

For the purpose of this paper, a research software tool has been developed to allow user-directed corruption of any selected relationship. The relationship types, such as finish-to-start (FS), start-to-start (SS), finish-to-finish (FF), or start-to-finish (SF) can be set to an illegal setting such as “xx”. Alternatively, the relationship lag can be optionally set to “NULL” (does not exist), which is completely different from a legal value such as zero (0).

Corrupt Relationship Type

The first trial of deliberate corruption involves corrupting the relationship type. Relationships # 2-6 are set to the illegal and nonsensical value of “xx”. The CPM is then recalculated. The resulting logical diagram looks like that in Figure 2 below.



Activity ID	Activity Name	Original Duration
⊟ Corrupt-2	Corrupt F	10
A100	Start	0
A110	Act 1	5
A120	Act 2	5
A130	Act 3	5
A140	Act 4	5
A150	Act 5	5
A160	Finish	0

Figure 2 – Corrupt Relationship Type CPM

P6 still displays relationships between all of the activities, but the graphical depiction is that of a FS relationship. This looks to be far different from the original baseline schedule in Figure 1.

P6 Schedule dialog box provides an option to record all scheduling results, such as warnings and errors, in a log file. Figure 3 shows an excerpt from such file after the CPM computations of the corrupt relationship types schedule is performed. Notice that there are no errors detected and the warnings are just standard information.

```
Statistics:
-----------
          # Projects....................................................1
          # Activities..................................................7
          # Not Started..................................................7
          # In Progress..................................................0
          # Completed....................................................0
          # Relationships................................................6
          # Activities with Constraint...................................0

Errors:
-------
Warnings:
---------
          Activities without predecessors.................................1
                    Project:  Corrupt-2          Activity:  A100     Start

          Activities without successors..................................1
                    Project:  Corrupt-2          Activity:  A160     Finish

          Out-of-sequence activities.....................................0

          Activities with Actual Dates > Data Date........................0

          Milestone Activities with invalid relationships.................0

          Finish milestone and predecessors have different calendars.......0
```

Figure 3 – Corrupt Relationship Types CPM Log

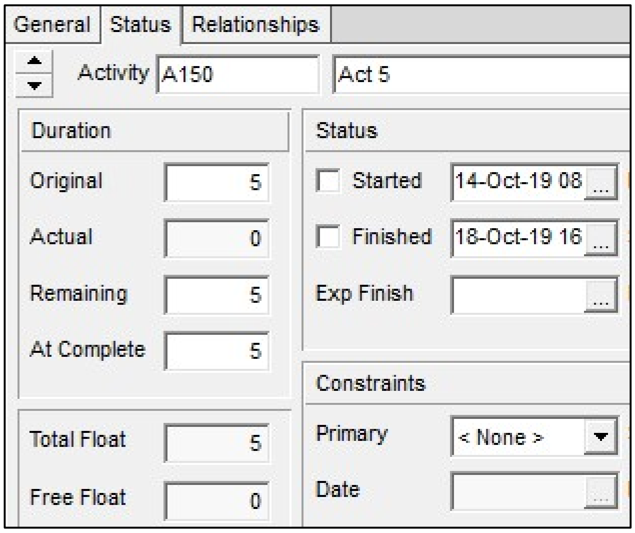
As a further example, Figure 4 shows a similar P6 CPM log report stemming from a calculation in which a CPM loop is discovered. Errors in the construction of the CPM schedule are supposed to be displayed in the Error section. The fact that no relationship types exist in the corrupt schedule should mean that the CPM software has no way to properly proceed. This fact should have been reported here just like CPM loops.

```
Statistics:
-----------
          # Projects......................................................1
          # Activities...................................................7
          # Not Started..................................................7
          # In Progress..................................................0
          # Completed....................................................0
          # Relationships................................................7
          # Activities with Constraint...................................0

Errors:
-------
          Schedule loops detected.........................................1

          Loop      Project               Activity        Description
          ---------------------------------------------------------------------------------
          1         Corrupt-1               A100            Start
                    Corrupt-1               A110            Act 1
                    Corrupt-1               A120            Act 2
                    Corrupt-1               A100            Start

Warnings:
---------
          Activities without predecessors.................................0

          Activities without successors...................................1
                    Project:  Corrupt-1          Activity:  A160      Finish

          Out-of-sequence activities......................................0

          Activities with Actual Dates > Data Date........................0

          Milestone Activities with invalid relationships.................0

          Finish milestone and predecessors have different calendars.......0
```

Figure 4 – CPM Log Schedule Loops Error Report

Another odd fact visible in the corrupt relationship type schedule is that none of the activities are shown in red. In this case, this color is used to denote that the activities have zero total float.

Figure 5 shows the status of activity A150. Even though the critical path must directly go through this activity, the figure clearly shows that the CPM calculated a total float of 5.



General | Status | Relationships

Activity A150 Act 5

Duration

Original 5

Actual 0

Remaining 5

At Complete 5

Total Float 5

Free Float 0

Status

☐ Started 14-Oct-19 08

☐ Finished 18-Oct-19 16

Exp Finish

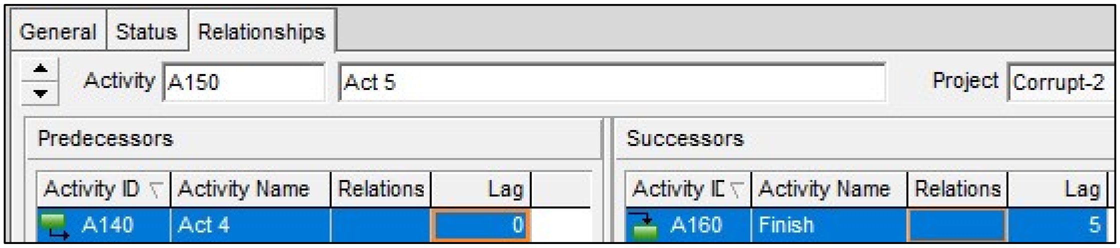
Constraints

Primary < None >

Date

Figure 5 – Corrupt Relationship Total Float

Finally, the expert user of P6 should learn to recognize what a relationship type corruption looks like using P6. Figure 6 displays the relationships to and from activity A150.



General | Status | Relationships

Activity A150 Act 5 Project Corrupt-2

Predecessors

Activity ID	Activity Name	Relations	Lag
A140	Act 4		0

Successors

Activity IC	Activity Name	Relations	Lag
A160	Finish		5

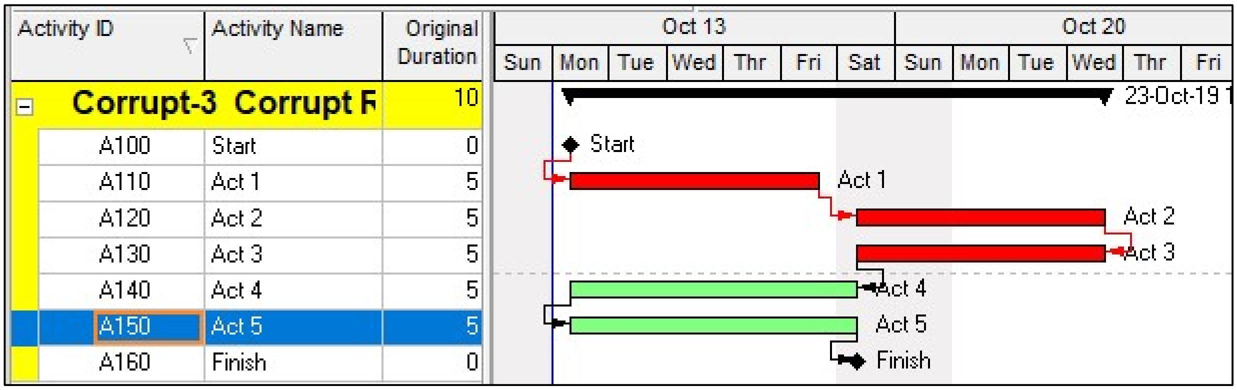
Figure 6 – Relationship Tab Showing Corrupt Relationship Type

The relations column shows the currently set relationship type. Legal values are FS, SS, FF, and SF. In both cases, the relationship type both to and from that activity are displayed as blank. The research tool has set the relationship value to “xx”, however P6 displays blank field instead.

Corrupt Relationship Lag

A second schedule is created from a copy of the baseline schedule to test how well P6 deals with corrupt relationship lags. Lags can be any number from negative to positive and zero. What is unacceptable in the database is if the lag value is missing. This is not the same thing as zero (0), which is a valid number. This condition of a missing value is called “NULL”.

The research tool has been utilized to set all of the relationship lags to NULL in the third schedule. The CPM is then recalculated, and the resulting network is displayed in Figure 7.



Activity ID	Activity Name	Original Duration
Corrupt-3	Corrupt F	10
A100	Start	0
A110	Act 1	5
A120	Act 2	5
A130	Act 3	5
A140	Act 4	5
A150	Act 5	5
A160	Finish	0

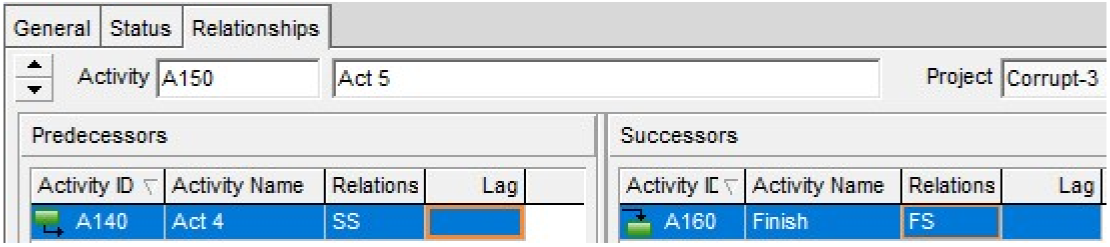
Figure 7 – CPM Schedule with "NULL" Lags

The only outward sign of a change is that activity A160 (finish) is now scheduled immediately after activity A150 instead of 5 days later. Evidently, the CPM calculation routine simply assumes that any missing lag value must be zero. The CPM calculation log from this schedule is excerpted in Figure 8 below. There is no indication that P6 noted anything wrong during the CPM calculation.

```
Statistics:
-----------
            # Projects......................................................1
            # Activities...................................................7
            # Not Started..................................................7
            # In Progress..................................................0
            # Completed....................................................0
            # Relationships...............................................6
            # Activities with Constraint...................................0

Errors:
-------
Warnings:
---------
            Activities without predecessors.................................1
                        Project:  Corrupt-3            Activity:  A100      Start

            Activities without successors..................................1
                        Project:  Corrupt-3            Activity:  A160      Finish

            Out-of-sequence activities......................................0

            Activities with Actual Dates > Data Date........................0

            Milestone Activities with invalid relationships.................0

            Finish milestone and predecessors have different calendars.......0
```

Figure 8 – Corrupt Relationship Lag Log Report

What should the P6 user look for as an indicator of such a relationship lag corruption? Figure 9 shows the relationship tab for activity A150. Lag fields for both the predecessor and the successor relationships are blank.



Activity: A150 | Act 5

Project: Corrupt-3

Predecessors

Activity ID	Activity Name	Relations	Lag
A140	Act 4	SS	

Successors

Activity ID	Activity Name	Relations	Lag
A160	Finish	FS	

Figure 9 – Relationship Tab for a Corrupt Relationship Lag

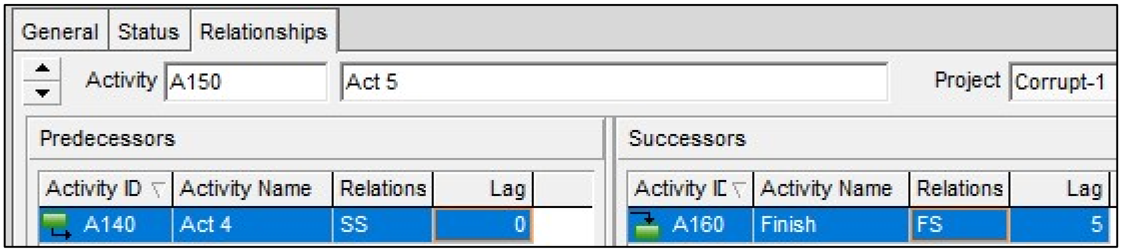
Repairing Corrupt Relationship Data

The solution for the P6 user to correct such corruptions is to research the correct data and manually update the information. This solution first requires the user to identify the corruptions and then to find the most recent source of non-corrupted data.

The P6 user can identify relationship corruptions by exporting the schedule relationship information to Microsoft Excel and then sorting by type and then by lag. Finding corrupt type information involves eliminating any relationship with correct settings and leaving those without. When exporting from P6, missing lags are sorted at the top of the column as missing any value.

Next, the P6 user should research the correct values to insert into the corrupted relationship data. A common value cannot be simply assumed, but must be carefully researched in order to fix the corruption. Start looking into recent CPM schedule submittals of earlier versions of this schedule until the non-corrupted data is found. At the very least, the baseline schedule should represent the planned values that are currently missing. Hopefully, the reviewer of the baseline schedule did not fail to notice corrupt relationships in the baseline schedule. This would be a significant review failure.

The final step in repairing corrupt P6 relationship data is to use P6 to open the corrupt schedule and display the relationships missing the correct data, such as shown earlier in Figures 6 and 9. Click the cursor in the blank relationship type or lag fields and type in the correct data. The corrected tab would now look like Figure 10 below.



Activity: A150 | Act 5

Project: Corrupt-1

Predecessors

Activity ID	Activity Name	Relations	Lag
A140	Act 4	SS	0

Successors

Activity ID	Activity Name	Relations	Lag
A160	Finish	FS	5

Figure 10 – Corrected Relationship Information

Reason for Some Corruptions

Database / Programming Mismatch

Some corruptions exist because the software rules do not line up with the database rules. A communication break between the database and P6 application may occur when the rules are enforced only by the database or only by the P6. For example, P6 requires project IDs to be unique, so that a user cannot create or change a project ID that is a duplicate of an existing one. The P6 database, however, does not enforce the project ID uniqueness rule. The definitions for database fields can be set to only allow unique entries and thus enforce uniqueness, but in this case it does not. Since the database does not enforce the project ID uniqueness rule, if some P6 software routine makes a mistake and allows a duplication, it is possible for the mistake to be permanently incorporated in the database with unknown consequences. This duplication ID error is a known occurrence in P6 database management. [1]

The same issue of mis-matched program and database rules is evident on other parts of the P6 database. The P6 program no longer allows a user to not enter a lag value when creating or modifying a relationship[1]. The P6 database does not enforce the same restriction on this field. Many existing P6 schedules today have this error in their databases.

Poor Safety Checks on Importing

Many errors in the P6 database are introduced by importing erroneous data. The xer import process is notorious for failing to check for the appropriateness of incoming data before storing it in the database. [2] The classic error is failing to strain out text that does not adhere to the standards for American Standard Code for Information Interchange (ASCII) text. ASCII is a code for representing one-hundred and twenty-eight (128) English characters as numbers, with each letter, number, or other signal assigned a number from zero (0) to one-hundred and twenty-seven (127). Characters greater than value “127” should not be in the input, as they cannot be displayed as text in the P6 application.

Other errors are introduced by mis-managing the imported data. For example, when risk category records are imported, duplicates do not appear to be rejected, but instead are added to the end of the existing and ever-expanding list.

Risk Category Errors

Newer versions of Primavera P6 contain a user-maintained risk assessment feature that can be applied and tracked on an activity-by-activity basis. The risks are enumerated in risk category field which is maintained on a project level. The user must enumerate and describe the risks before being able to assign them to individual activities.

[1] Prior to P6 Version 8.4, the P6 user could enter a blank (NULL) value for the relationship lag without generating an error. This action is not allowed by software for all newer versions of P6.

The issue with maintaining such project level risks is that the list of risks often grows with each project update. It appears that the update adds risks to the existing risks, instead of only adding the newly added ones. In addition, this list is subject to corruption where hundreds or even thousands of new lines are added to the risk type list. These corrupt lines of text quite often only list what to the user would be seen as gibberish. A typical example of such corrupt lines is included in Figure 11.

```
Èçìåíåíèå óñëîâèé, ñðîêîâ
TrÃ¿Â¿Ã¿Â¿Ã¿Â¿Ã¿Â¨s Bas
M?Â©dium
Inovaï¿½ï¿½o e Integraï¿½ï¿½o
```

Figure 11 – Corrupted Risk Type Example

The odd characters are created because normal text is limited to character numbers 32 to 255, while these corrupted entries can be of any value way beyond the normal text set boundaries. These odd characters can tie-up P6 as it is loading the schedule and prevent the user from opening the schedule.

This issue has been resolved with the fix included in Primavera P6 patch set 18.8.4.1, as well as all subsequent patch sets. [3] This fix prevents P6 application from crashing when it opens a corrupted risk type schedule. However, the above referenced article does not mention whether the database is cleaned of these odd entries, nor does it say if the fix prevents the continual addition of new, unwanted risk type entries. Initial investigations conducted by the authors indicate that the risk type database clean-up is not carried out.

The user may have this risk category corruption without it causing P6 to crash when opening the schedule. The database will continue to grow larger, containing these unwanted entries. This results in a slow opening time of the schedule and greatly increases the creation/load times of an xer schedule backup file that contains extra risk category entries. Oracle support article suggest removing corrupted risk categories via the following workaround:

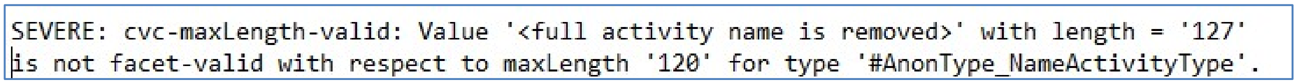
1. Login to P6 Professional as an admin superuser.
2. Navigate to admin > admin categories.
3. Select risk categories.
4. Scroll through the list of risk categories and delete the ones with the invalid characters.

This workaround can be an extensive process, because some corrupted databases may contain several thousand invalid risk category entries and manual deletion process handles only one entry at a time. The authors of this paper offer the alternative solutions in sections ‘Helpful Database Queries’ and ‘Preventative Strategies’.

Non-Extendable Field Lengths

Some field lengths in P6 are defined in the database and cannot be extended beyond a hard-coded maximum value. For example, the activity name field has the maximum allowable length of one-hundred and twenty (120) characters. This size is not extendable and P6 users who want to propose the change are directed to submit an enhancement request [4].

The problem occurs when project files are exported and imported between different scheduling applications where the second application does not have the same field length restrictions. Exchange of project files between Microsoft Project and P6 is a good case in point. Import of Microsoft Project xml file with the length of activity names exceeding the limit set in P6 database results in import failure. The log file assigns 'severe' level to the error as illustrated in Figure 12. The P6 user is impacted by not being able to import the project schedule into P6 application. In the context of this paper the project is corrupted.



```
SEVERE: cvc-maxLength-valid: Value '<full activity name is removed>' with length = '127'
is not facet-valid with respect to maxLength '120' for type '#AnonType_NameActivityType'.
```

Figure 12 – Failed Import Log File

P6 versions prior to release 15.2.4.0 can handle the truncation of the field values in the SQL Server. [5] The workaround for P6 users of later releases is to import the corrupted Microsoft Project xml file into SQLite database and manually truncate any field exceeding maximum allowable length. That is why a separate section on how to install and upgrade a local SQLite database is presented later in this paper.

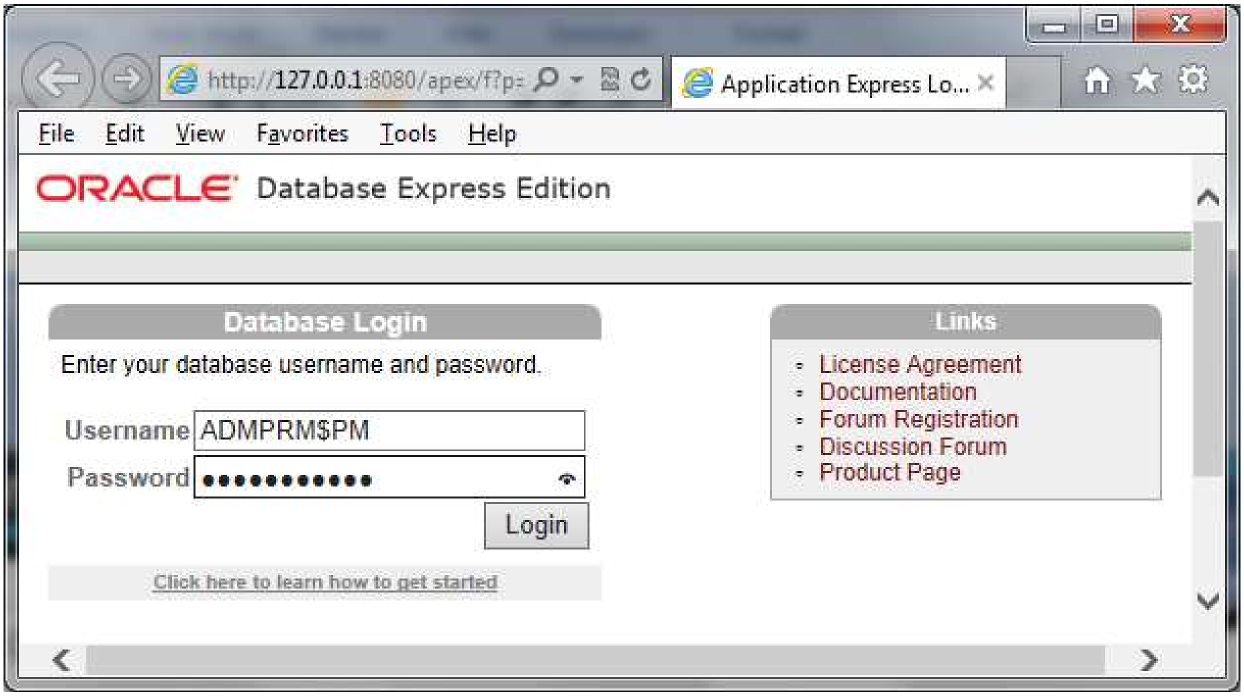
Helpful Database Queries

Some database corruption issues can only be discovered and resolved by bypassing the P6 interface and dealing directly with the underlying database manager. This typically falls in the domain of the information technology (IT) team, who have database administrator credentials and are experienced in using the structured query language (SQL). A command from the IT programmer to the database manager software is called a query.

There are a number of custom queries that help to identify duplicate or orphan records, as well as keep P6 databases clean by purging deleted records, expired user sessions and so on. These queries must be run using database SQL servers – either Oracle or Microsoft. The identified data errors can either be removed directly from the database via custom queries or can be fixed manually in the P6 client application. An alternative solution of using 3rd party applications is described in 'Add-on Software Products' section of this paper.

How to run a Query in Oracle Database (Express)

Browse to the Windows system function start, programs, Oracle Database 10g Express Edition, Go To Database Home Page. The resulting home page is displayed in Figure 13 below. At the login page enter the database account information. For example, one of the default usernames for standalone installation is ADMPRM$PM:


http://127.0.0.1:8080/apex/f?p=
Application Express Lo...
File Edit View Favorites Tools Help
ORACLE Database Express Edition
Database Login
Links
License Agreement
Documentation
Forum Registration
Discussion Forum
Product Page
Login
Click here to learn how to get started

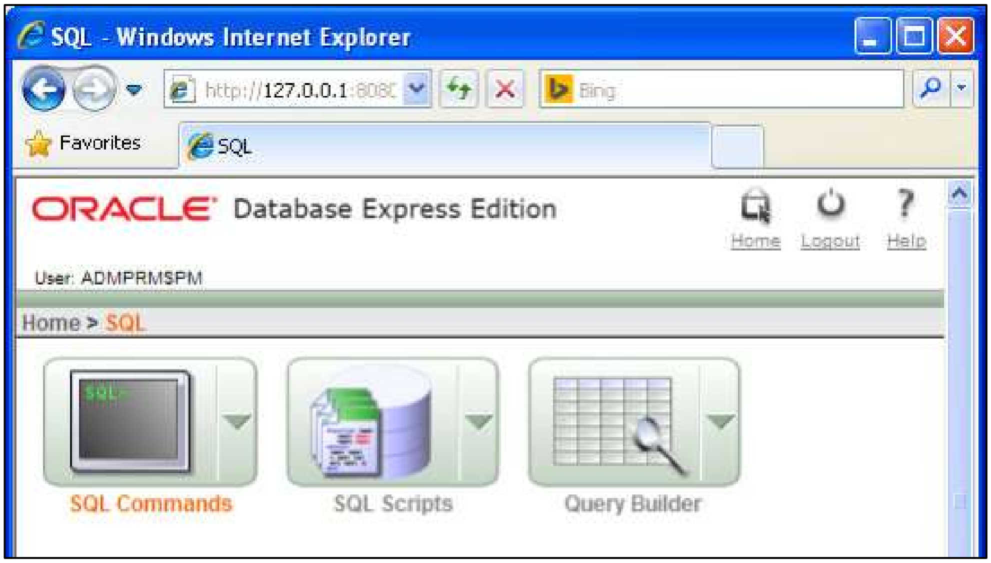
Enter your database username and password.

Username ADMPRM$PM

Password ●●●●●●●●●●●●

Figure 13 – Oracle Database Express Edition Login Page

Select 'SQL' button, then 'SQL Commands' button shown in Figure 14.



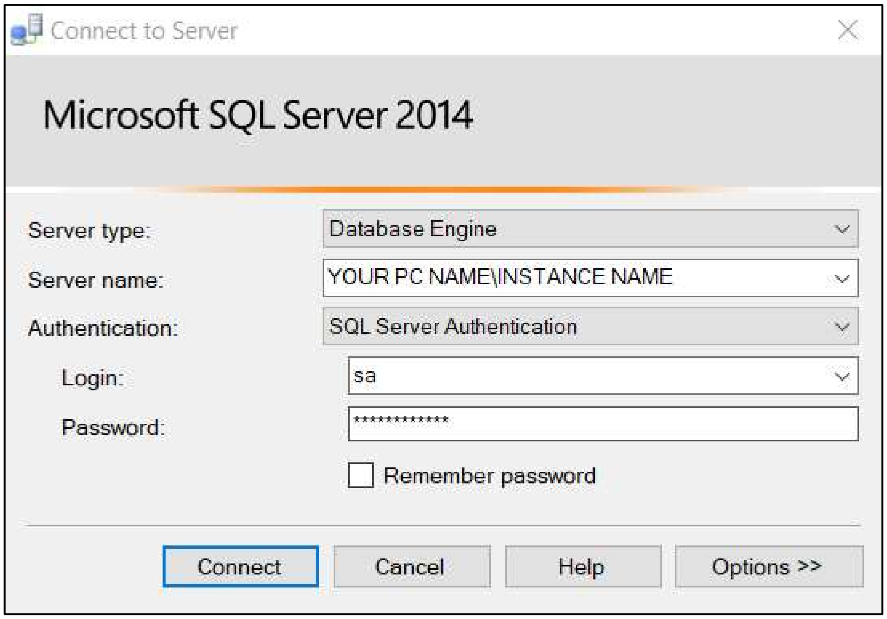
User: ADMPRM$PM

Figure 14 – Oracle Database Express Edition SQL Commands Option

A new query can be entered in the window. Select ‘Run’ to execute. [6]

How to run a Query in Microsoft SQL Server (Express)

Browse the Windows system command to start, programs, Microsoft SQL Server 20xx, SQL Server 20xx Management Studio. At the login page enter server and instance name. For example, the standalone installation with default settings would have server name as ‘localhost’ and instance name as ‘primavera’. Set ‘Authentication’ and ‘Login’ as shown in Figure 15. Type the password as configured during the standalone installation and click ‘Connect’:


Connect to Server

Microsoft SQL Server 2014

Server type: Database Engine

Server name: YOUR PC NAME\INSTANCE NAME

Authentication: SQL Server Authentication

Login: sa

Password: ************

□ Remember password

Figure 15 – Microsoft SQL Server Login Page

In the left pane, under object explorer, expand the ‘Databases’ folder. Highlight and right click the database to run a query against and select ‘New Query’. (Figure 16) A new query can be entered in the blank window of the middle task pane. [7]

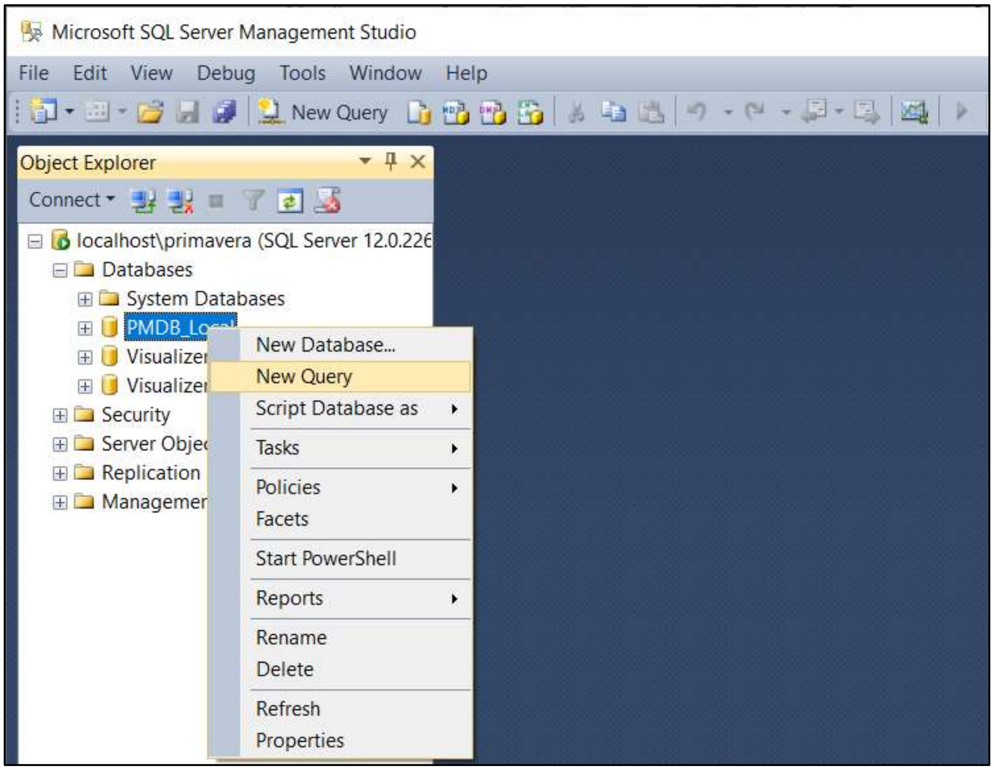


Figure 16 – Microsoft SQL Server Object Explorer Window

Remove Invalid Risk Type Records

The following command can be used to purge the invalid risk categories records directly from the corrupted database. It is recommended to back-up database before running this query:

```
DELETE        from RISKTYPE where risk_type != asciistr(risk_type)
```

This SQL command fix does not eliminate any duplicate or unwanted records that do not contain corrupted text. That would have to be done through the 4-step manual process as shown in previous section called 'Risk Category Errors'. Automated fixes are addressed in a later section called 'Add-on Software Products'.

Initialize / Re-start Background Jobs

Cleanup database tasks, such as purging deleted records or expired user sessions, are handled in P6 by background jobs called SYMON (System Monitor) and DAMON (Data Monitor). Both of these processes are pre-configured and should start running automatically once the P6 Client is installed with SQL Server – Microsoft or Oracle. [10]

P6 user can check what jobs exist on the SQL Server database by running a query with sysdba privileges in Oracle Database Express:

```
SELECT      JOB, LOG_USER, LAST_DATE, NEXT_DATE, WHAT from DBA_JOBS
```

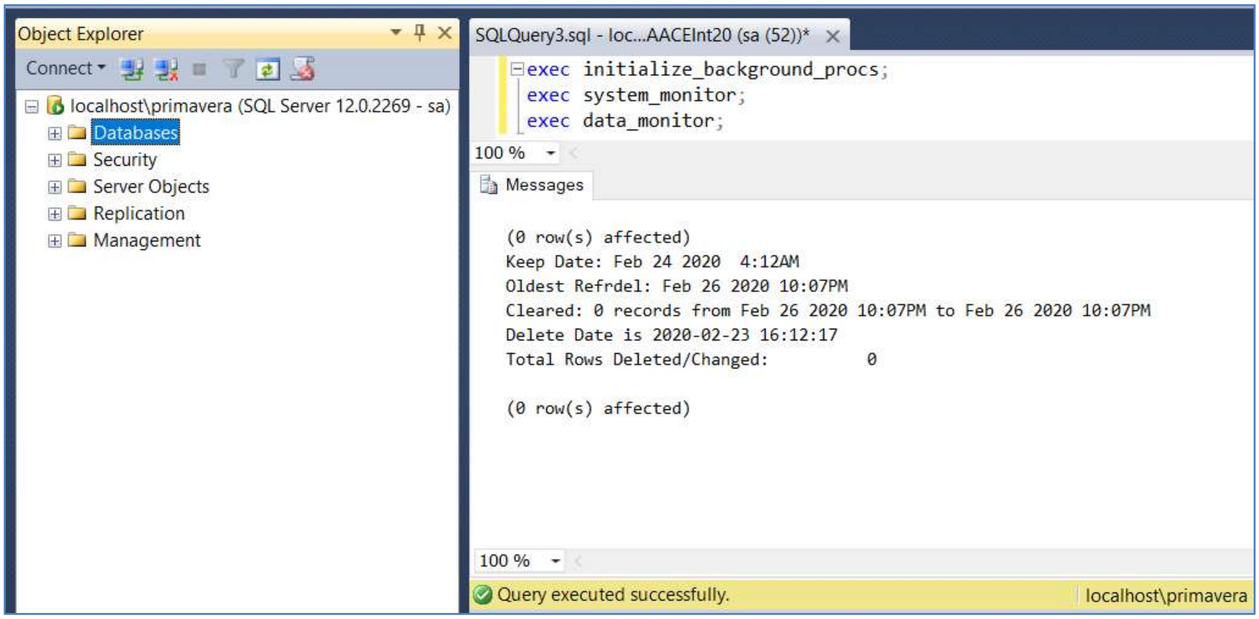
Or with sa privileges in Microsoft SQL Server:

```
SELECT      * from msdb.dbo.sysjobs
```

If the P6 background processes are not listed, they can be reinitialized by running the following commands against local database:

```
EXEC        initialize_background_procs
EXEC        system_monitor
EXEC        data_monitor
```

Oracle knowledgebase also lists queries that check the background jobs DAMON and SYMON exists and show job history as illustrated in Figure 17. [10]


Object Explorer
Connect
localhost\primavera (SQL Server 12.0.2269 - sa)
Databases
Security
Server Objects
Replication
Management
SQLQuery3.sql - loc...AACEInt20 (sa (52))*
100 %
Messages
100 %
Query executed successfully.
localhost\primavera

```
exec initialize_background_procs;
exec system_monitor;
exec data_monitor;
```

```
(0 row(s) affected)
Keep Date: Feb 24 2020  4:12AM
Oldest Refrdel: Feb 26 2020 10:07PM
Cleared: 0 records from Feb 26 2020 10:07PM to Feb 26 2020 10:07PM
Delete Date is 2020-02-23 16:12:17
Total Rows Deleted/Changed:         0

(0 row(s) affected)
```

Figure 17 – Primavera Job Processes

Preventative Strategies

The authors of this technical paper stay firmly behind their recommendation to have at least two P6 databases at user's disposal: one for working with production data, and one as a 'testing/cleansing' space. [2] Yet, oftentimes it is logistically challenging to have more than one SQL Server database installed on user's machine, especially if patching and upgrading Primavera P6 is undertaken by IT personnel. Thus, an option to have SQLite database is presented below. Another preventative strategy that falls under database management's best practice recommendation is automating the database backup process. This allows users to restore P6 data from the last backup if data corruption occurs. Finally, having a 3rd party add-on applications can relieve P6 users from having to manually run SQL queries, and can potentially repair data corruption at one click of a button.

SQLite Manual Install

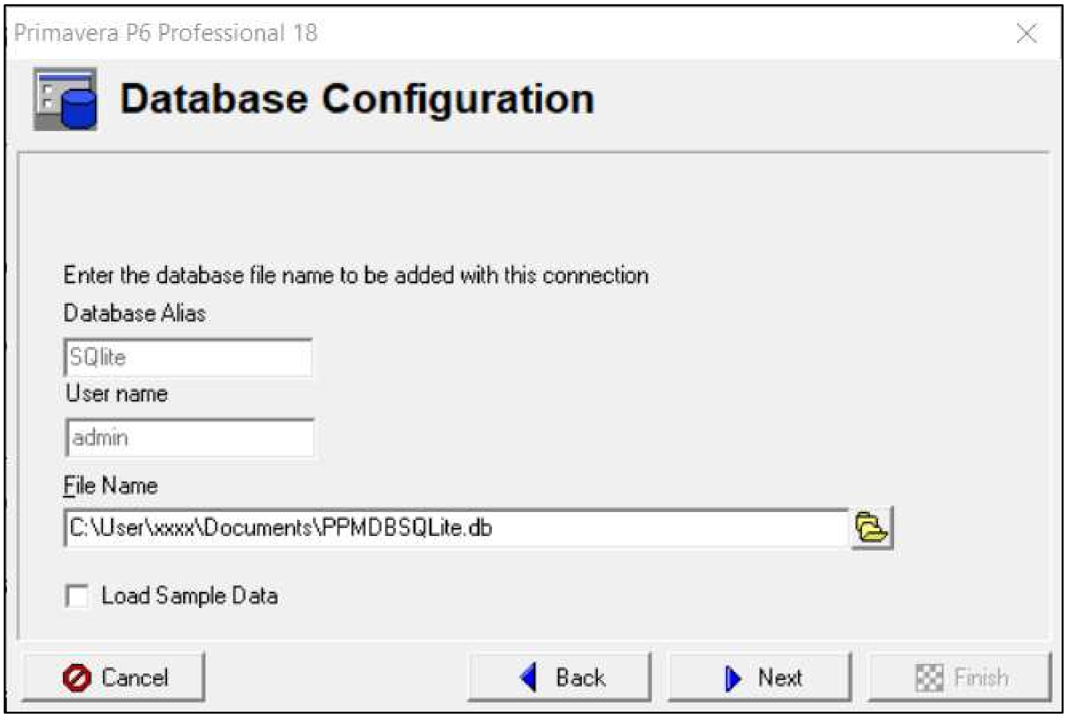
SQLite file is a serverless, transactional, self-contained and free database engine. This option has been introduced as a standalone installation of the P6 client since Primavera release 8.4 in October of 2014. Installing and upgrading a SQLite P6 database is not a complicated process and can be performed by a P6 user without professional assistance of IT personnel using the steps and references outlined below.

There are a few considerations that need to be taken into the account prior to installation and usage of P6 SQLite database, such as:

- P6 installation files must be available on the local drive or the workstation where P6 is installed, not on a network or shared drive.
- SQLite can be installed on the same workstation as Microsoft SQL Server Express.
- Potential size of SQLite database file is a hundred and forty (140) terabytes; however, it is strongly recommended that only one user accesses it. In other words, SQLite is not designed for multiple and concurrent P6 users.
- There are multiple interface changes and limitations with SQLite database related to its single-user nature [8].

To install a P6 standalone SQLite database, the user should follow the steps outlined below:

1. Launch P6 client, then click the ellipses next to database and select add.
2. On the database configuration screen enter the name for the database alias (which is any meaningful name) and select ‘P6 Pro Standalone (SQLite)’ as driver type.
3. Choose ‘Add a new standalone database and connection’ radio button.
4. Enter the password for the admin user.
5. Accept default settings or customize location and name of the database file [Fig. 18]



Primavera P6 Professional 18

Database Configuration

Enter the database file name to be added with this connection

Database Alias: SQlite

User name: admin

File Name: C:\User\xxxx\Documents\PPMDBSQLite.db

☐ Load Sample Data

Figure 18 – Standalone SQLite database configuration

When a successful connection to the newly created database is established, the user can return to the P6 application and login into the SQLite database.

To upgrade an existing SQLite database, first upgrade the P6 client, then follow the steps below:

1. Launch the upgraded P6 client, click the ellipses next to database and select configure.
2. On the database configuration screen, enter the name for the database alias (which is any meaningful name) and select ‘P6 Pro Standalone (SQLite)’ as driver type.
3. Choose ‘Add a connection to an existing standalone database’ radio button.
4. Browse to the SQLite database file’s location on the computer, click next.

When a successful connection message is displayed, click finish to return to the P6 application. At the P6 login screen select SQLite database alias from Step 2 above, enter the password and click connect. Select OK on a database version mismatch message. Next prompt asks if the user wants to upgrade the SQLite database to latest schema – click ‘Yes’. [9]

Add-on Software Products

There are two issues involved in addressing the risk type corruption: eliminating the problem in the P6 database and preventing a reoccurrence of the problem by eliminating the problem in the import file. The first fix would involve SQL queries. The second would programmatically review and edit the xer or xml import text files. There is one third-party software that conditions xer and P6 xml files to remove the risk type corruption prior to importing. [11]

An alternative process to better eliminate only the needed risk types would be to identify all risk types actively assigned to existing activities in the project and then to eliminate any other risk type entry associated with that project. This is a safer process as risk types are not global so that other schedules will not be affected. This is the process employed by one third-party software fix [12].

Conclusion

This paper is a third effort by the authors to bring attention of P6 users to an importance of recognizing various types of data corruptions within the application, as well as being able to fix these corruptions. The practical manifestations of what a corrupt P6 schedule looks like are presented and proper procedures on how to repair it are offered. Issues such as scheduling errors are illustrated and why they matter to the average scheduler are explained.

The integrity of the schedule can be compromised due to corrupted data and this paper outlines the ways and means available to P6 users to address various corruption scenarios. Some of the issues described can be avoided or mitigated by simply having a testing SQLite database installed and used as a first line of defense against corrupted data in scheduling files. Another option utilizes a third-party software that flags, and fixes compromised data at a click of a button.

Preventative strategies and helpful database queries listed in the paper are solid and tested ways of addressing P6 data corruption and, for the most part, do not require the assistance from a qualified IT professional. The authors hope to empower P6 users with this research and improve their experience of working with P6 scheduling application.

References

[1] P6 File Corruption, Part 2, 2017 AACE International Technical Paper, Ronald M. Winter PSP FAACE and Marina G. Sominsky PSP EIT PMP

[2] P6 File Corruption, 2016 AACE International Technical Paper, Ronald M. Winter PSP FAACE and Marina G. Sominsky PSP EIT PMP

[3] My Oracle Support, Error 'TCVirtualTable.OptFieldIndexOF:VTable is not loaded - PROJECT' When Attempting to Login to P6 Professional Using CloudConnect (Doc ID 2298125.1)

[4] My Oracle Support, Can the Activity Name Field Length Be Increased From the Current Maximum of 120 Characters? (Doc ID 2487064.1)
04-Dec-2019 (Last Update)

[5] My Oracle Support, "INSERT statement conflicted with the FOREIGN KEY constraint "fk_projwbs_project". The conflict occurred in database "PMDB", table "dbo.PROJECT", column 'proj_id'." When Importing Primavera Risk Analysis XER File into P6 Running on SQL Server. (Doc ID 2186368.1) 06-Mar-2019 (Last Update)

[6] My Oracle Support, How to Run a Query/Command/Script in Oracle Database 10g Express Edition (XE) (Doc ID 982225.1) 25-Sep-2019 (Last Update)

[7] My Oracle Support, How to Run a Query Using SQL Management Studio (Express) (Doc ID 911508.1) 09-Aug-2019 (Last Update)

[8] My Oracle Support, What are the Limitations With P6 Client Standalone Installation and SQLite? (Doc ID 1946536.1) 04-Dec-2019 (Last Update)

[9] My Oracle Support, How to Upgrade a SQLite Database When Upgrading P6 Professional to 15.1 or Later (Doc ID 2087187.1) 04-Dec-2019 (Last Update)

[10] My Oracle Support, Error: "The Database Has Not Been Configured To Run Background Jobs." When Logging Into P6 Project Management (Doc ID 905203.1) 25-Jul-2019

[11] Schedule Analyzer for the Enterprise, Purge POBS Module by Ron Winter Consulting LLC.

[12] Schedule Analyzer Enterprise Forensic by Ron Winter Consulting LLC,
http://scheduleanalyzer.com/eforensic_brochure.htm