**OPIM 5272 –**

**Data Management and Business Process Modeling – Bug Management in Gaming Studio – Project Phase 2**

**GROUP 4**

* **PRASHANT GAIKWAD**
* **TUSHAR GHULE**
* **RUCHIKA PATTJOSHI**

# **REPORT REQUIREMENT FOR BUSINESS**

1. **Query Report-1**

**Business Case:**

For any bug that was reported, once the bug is fixed, how the fix was made should be documented by the employee who worked on it. The document should include details of the bug resolution date, whether the document has been created or not, if not created then how many days has it been since the bug was fixed.

**Benefits:**

By having this report, the employee who worked on the bug fix can be notified to create a bug fix documentation to have records on how it was fixed, how long did it take to fix the bug, what was the severity of the bug, how many users were impacted. By doing so, if a similar bug is reported in the future, then this document can be referred for a quick resolution.

**Report Query Question:**

Give a report that shows the bug list for which the bug fix document was not created post the bug was fixed. Report should display the bug resolved to date, right next to it display how many days has it been since the resolved date and the current date. Also, display the employee who worked on it (emp id, concat first name, last name with space in between), what is the bug Id, severity of the bug. Display the output only if there is no documentation available. Order the output by ascending order by BugID.

**SQL Query:**

Select B.Bug\_Id, B.Bug\_Severity, DM.Emp\_ID,

(E.Emp\_First\_Name || ' ' || E.Emp\_Last\_Name) AS "Employee\_Name",

BM.BUG\_RESOLVED\_DATE,

abs(trunc(sysdate) - BM.BUG\_RESOLVED\_DATE) AS "No of Days since Bug resolved"

from **RUP21001.**bug\_details B

join **RUP21001.**bug\_management BM

on B.Bug\_ID=BM.Bug\_ID

join **RUP21001.**bug\_fix\_document BD

on B.Bug\_ID = BD.Bug\_ID

join **RUP21001.**Document\_Management DM

on BD.Bug\_Doc\_ID = DM.Bug\_Doc\_ID

join **RUP21001.**Emp\_Details E

on DM.Emp\_ID=E.Emp\_ID

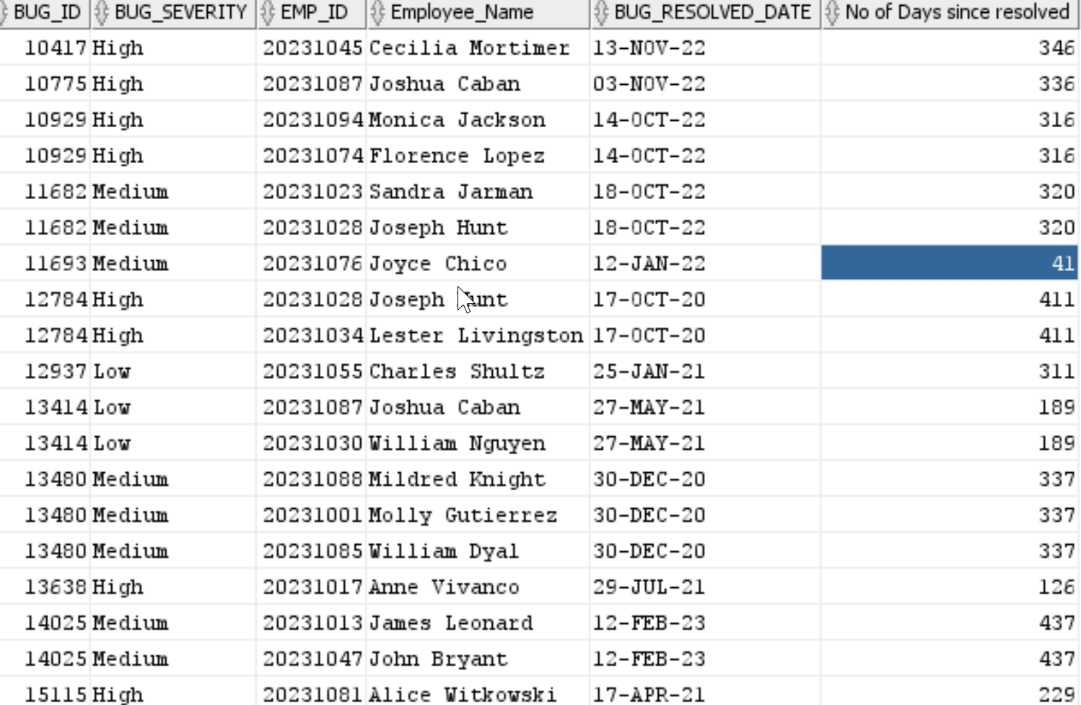
where DM.Doc\_Created\_Date is not Null

and B.Bug\_Status='Resolved'

order by B.Bug\_ID

;

**Output :**





1. **Query Report-2**

**Business Case:**

There are 9 distinct types of bugs based upon where are they generated in the game. For example, Save Glitch, In-Game Purchase, Crashing, Graphics, and so on. A manager would like to know which type of bug is being created the greatest number of times and what’s the avg fix time (resolve date - report date) for the same.

**Benefits:**

By knowing what type of bugs generate the greatest number of bugs and consume most work hours, a manager can reassign the bug severity to these types and allocate resources accordingly. Usually, the development team is divided according to the types of bugs they work upon. So, the relocation can include assigning a senior programmer to the type of the most severe bug.

**Report Query Question:**

Give a report that will give bug fix duration (resolved - reported) and the count of the bugs for each bug type.

**SQL Query:**

Select BD.bug\_type, trunc(Avg(BM.bug\_resolved\_date - BM.bug\_report\_date)) as Avg\_Bug\_Fix\_Duration, count(\*) as No\_Of\_bugs

from **RUP21001.**bug\_management BM

join **RUP21001.**bug\_details BD

on BM.bug\_id = BD.bug\_id

group by bd.bug\_type

order by Avg\_Bug\_Fix\_Duration, No\_Of\_bugs desc

;

**Output:**

Graphical user interface, application

Description automatically generated



1. **Query Report-3**

**Business Case:**The Project Manager would like to know for each game how many packages have been created for the bug reported.

**Benefits:**This will help to understand which game is more active depending on how many packages have been created for the game, the higher the count that means requires more system engineer employees are required for that game.

**Report Query Question:**Create a report that shows a count of packages created for each game, order the report by a count of packages in descending order.

**SQL Query:**

select gd.game\_name, count(gpd.package\_id) Count\_of\_Packages from RUP21001.game\_package\_details GPD

join RUP21001.Bug\_Details BD

on GPD.package\_id = bd.package\_id

join RUP21001.Game\_Details GD

on bd.game\_id = gd.game\_id

group by gd.game\_name

order by count\_of\_packages desc

;

**Output:**



**Table

Description automatically generated**

1. **Query-Report-4**

**Business Case:** For the annual employee evaluation, the Project Manager is trying to find ways in which he could measure the employee performance. He observes that there is a difference between the number of bugs resolved by each employee and that could help to provide the potential annual bonus too.

**Benefits:** Employees can be evaluated without any bias and the annual bug resolution performances can be shared with each individual employee. The bonus based on the bugs will help motivate the employees to improve their productivity and resolution rate

**Report Query Question:** Create a report showing the developer and the number of bugs they have resolved. The project manager has decided to award the extra bonus for employees that will be added to their net salary as follows –

1. If bug resolved count is between 1 to 10 – provide a bonus of $150 per bug resolved
2. If bug resolved count is between 11 to 20 – provide a bonus of $250 per bug resolved
3. If bug resolved count is between 21 to 30 – provide a bonus of $350 per bug resolved
4. If bug resolved count is greater than 31 – provide a bonus of $500 per bug resolved

Display the First Name, Last Name, Salary, Count of Bugs, and Bonus

**SQL Query:**

**SELECT**

**EMP\_ID, EMP\_FIRST\_NAME AS FIRST\_NAME, EMP\_LAST\_NAME AS LAST\_NAME, TO\_CHAR(EMP\_SALARY, '$999,999.00') AS ANNUAL\_SALARY, BUG\_COUNT,**

**(CASE**

**WHEN BUG\_COUNT <10 AND BUG\_COUNT > 0 THEN TO\_CHAR(BUG\_COUNT\*150, '$999,999.00')**

**WHEN BUG\_COUNT <21 AND BUG\_COUNT > 10 THEN TO\_CHAR(BUG\_COUNT\*150, '$999,999.00')**

**WHEN BUG\_COUNT <31 AND BUG\_COUNT > 20 THEN TO\_CHAR(BUG\_COUNT\*150, '$999,999.00')**

**ELSE TO\_CHAR(BUG\_COUNT\*150, '$999,999.00') END) AS "EMPLOYEE BONUS"**

**FROM (**

**SELECT**

**D.EMP\_FIRST\_NAME, D.EMP\_LAST\_NAME, D.EMP\_SALARY, B.EMP\_ID, COUNT(BUG\_ID) BUG\_COUNT**

**FROM RUP21001.BUG\_MANAGEMENT B**

**LEFT JOIN RUP21001.EMP\_DETAILS D**

**ON B.EMP\_ID = D.EMP\_ID**

**GROUP BY D.EMP\_FIRST\_NAME, D.EMP\_LAST\_NAME, D.EMP\_SALARY, B.EMP\_ID**

**)**

**ORDER BY BUG\_COUNT DESC;**

**OUTPUT:**

****

**Table

Description automatically generated**

1. **Query Report-5**

**Business Case:**

For any bug that was reported for a game, the company needs to keep a track of the progress being made in the bug fix, there are several kinds of status being maintained for the bug pix progress like resolved, new, committed, etc. So, the Department Head or top-level manager requests the status of progress for each game.

**Benefits:**

By having this report, the company can maintain records of the status of the work, how long does it take to resolve the bugs reported, what escalations can be made to stakeholders that fixing a bug for a particular game would take X number of days.

**Report Query Question:**

Give a report that shows the bug status for all the bug statuses available, and if a bug doesn’t have a status type display it as ‘Null’ for each game. Provide the count of each bug status for each of the games the company handles.

**SQL Query:**

select gd.game\_name, bug\_status,

(

case when bug\_status = 'Commited' Then count(bug\_status)

when bug\_status = 'Resolved' Then count(bug\_status)

when bug\_status = 'Removed' Then count(bug\_status)

when bug\_status = 'New' Then count(bug\_status)

when bug\_status = 'Approved' Then count(bug\_status)

else 0

end

) as count

from bug\_details BD join game\_details GD

on bd.game\_id = gd.game\_id

group by bug\_status, gd.game\_name

order by gd.game\_name

;

**OUTPUT:**

**Table

Description automatically generated**

