**BUSINESS QUESTION 1**: Ref BR5. Which proposals have gone over their proposed hours?

SHOW: Proposal name and proposed hours from table proposals, and the

sum of actual hours from table proposal progress as ‘total hours’.

JOIN: Join table proposals and table proposal progress based on the

proposal ID.

ORGANIZE: Group the results by proposal name then proposed hours and have

the calculation of ‘total hours’ be greater than the proposal’s

proposed hours.

**BUSINESS QUESTION 2**: Ref BR5. What percentage of proposed hours have been used up for

each proposal?

SHOW: Proposal Name, Percent hours used (sum of hours/proposed hours)

made into an integer for groupings, proposed hours, manager’s first

and last name concatenated.

CATEGORIZE: Categorize the percent of proposed hours with Red >= 100%, Yellow

>= 60%, Green < 60%. For percent of hours used, name this as

Proposal Status. Take data from the proposals table.

JOIN: Join table proposals and table proposal progress based on Proposal

ID. Join table proposals and table project managers based on

Manager ID, where data from table proposals is matched with data

in table project managers. If there are no matches, table project

managers entries will be null. Join Table people with table project

managers based on manager ID being equal to people ID. If there

are no matches, table people entries will be null.

ORGANIZE: By proposal name, then by proposed hours, followed by employee

first and last name.

**BUSINESS QUESTION 3**: Ref BR4. What is the current total cost of each project?

SHOW: Project name from table projects, and the product of project charge

rate from table projects multiplied by total hours recorded. The

product shall be named ‘total cost’.

LEFT JOIN: To calculate total hours, take the sum of actual hours from table

proposal progress and also select project ID from table time entries.

These two tables will be joined based on the time entry ID and

grouped by the time entry ID. This result is known as table ‘th’ and

will be joined to table projects based on project ID, where data from

table projects are matched with data in table ‘th’. If there are no

matches, table ‘th’ entries will be null.

ORGANIZE: Order by ‘total cost’ from highest cost to lowest.

**BUSINESS QUESTION 4**: Ref BR4. Which projects are guided by proposals that are within the

proposed hours?

SHOW: Project Name and Departments from table projects. Proposed hours,

Total Hours and First and Last name Concatenated (called Project

Manager name) from table proposal projects.

JOIN: Table Proposal Projects with Table projects based on Project ID. Join Tables Propsals and Proposal Progress based on proposal ID, show

Proposal ID, Approving Manager, Proposed Hours, and sum of

actual hours (shown as total hours). Group the joining of these

tables by proposal ID, proposed hours, and approving manager

where the sum of the actual hours is less than the proposed hours

for the project. Join the table proposalprogress with proposalprojects

on proposal ID.

LEFT JOINS: Join Table Project Managers and Table proposal progress but

return all rows from table project managers and only matching

values from table proposal progress based on proposal ID. Join

table People and Table Project Managers but return all rows from

table People and only matching values from table Project Managers

based on people ID.

**BUSINESS QUESTION 5**: Ref BR3. Which costly projects (Top 2) are guided by proposals that

are past their planned due dates?

SHOW: The top two project names from table projects, the ‘total cost’, which

is calculated by multiplying the charge rate from table projects by the

total hours from table ‘th’, the proposal name from table ‘due’, and

the project due date from table ‘due’.

JOIN: Join table projects and table proposal progress based on projects ID.

Join table proposal progress and table ‘due’ based on proposal ID.

Table ‘due’ is found in section SUBQUERY 2.

LEFT JOIN: Join table projects with table ‘th’ based on project ID, where data

from table projects are matched with data in table ‘th’. If there are no

matches, table ‘th’ entries will be null. Table ‘th’ is found in section

SUBQUERY 1.

SUBQUERY 1: Show the sum of actual hours from table proposal progress as ‘total

hours’ and the project ID from table time entries. Join table proposal

progress with table time entries based on time entry ID. Group this

table’s results based on project ID. Name this table ‘th’.

SUBQUERY 2: Show proposal ID, proposal name, and project due date from table

proposals. Join table proposals with table proposal progress based

on proposal ID. Join table proposal progress with table time entries

based on time entry ID. Group results by proposal ID, then project

due date, followed by proposal name having the greatest current

date (meaning today’s date) be before the resulting project’s due

dates.

ORGANIZE: Order by ‘total cost’ starting from greatest cost to lowest.

**BUSINESS QUESTION 6**: Ref BR4. What is the duration in days of each proposal (difference

between approval date and the latest time entry)?

SHOW: Proposal Name from proposal table, Amount of days between

acceptance date of proposal and last time listed as durationindays,

combine employee first and last name and call the return value

Project Manager. Phone number from people table listed as

ManagerContactNo. Company Name from client companies

JOIN: Proposal Progress and Table Proposals on Proposal ID. This is

achieved by selecting proposal id, taking the highest current date and

calling it “Latest Entry” from TableTime Entries, joining table proposal

progress with table time entries on time entry ID, and group these

values by Proposal ID. Call whole join “pp(short for proposal

progress)” and join based on proposal ID from table Proposals and

Table Proposal Progress.

LEFT JOIN: Join Table Project Managers and Table proposals but

return all rows from table proposals and only matching

values from table project managers based on Approving Manager

and manager ID respectively .Join table People and Table

Proposals but return all rows from table Proposals and only

matching values from table People based on manager id and

people ID respectively. Join table client companies with table

Proposals but return all rows from table proposals and only

matching values from table Client companies on Company ID.

**BUSINESS QUESTION 7:** Ref BR3. What is the track record of each company in terms of being

involved with proposals that are completed within the proposed

hours?

SHOW: Company name, customer phone number, customer city, and

customer state from table client companies. Show a count of

proposal IDs from table proposals as ‘Number of Proposals’. Show

the formula where a count of proposal IDs from table proposals

minus a count of distinct proposal IDs from table ‘PM’ is divided by a

count of proposal IDs from table proposals. This result is multiplied

by 100 and is casted as an integer and named ‘Percent Proposals

Within Hrs’.

LEFT JOINS: Join table client companies with table proposals based on company

ID, where data from table client companies are matched with data in

table proposals. If there are no matches, table proposals entries will

be null. Join table client companies with table proposals based on

company ID, where data from table client companies are matched

with data in table ‘PM’. If there are no matches, table ‘PM’ entries will

be null.Table ‘PM’ is found in section SUBQUERY 1.

SUBQUERY 1: Show the sum of actual hours as ‘total hours’ from table proposal

progress, and show company ID and proposal ID from table

proposals. Join table proposals with table proposal progress based

on proposal ID. Group by company ID, proposed hours, then

proposal ID, having ‘total hours’ be greater than proposed hours from

table proposals.

ORGANIZE: Group by company name, customer phone number, city, then state

from table client companies. Order by ‘Percent Proposals Within Hrs’

with results descending in value.

**BUSINESS QUESTION 8**: Ref BR3. What is the track record of each project manager in terms

of managing proposals that are within the proposed hours?

SHOW: Employee First and Last name combined and listed as “Project

Manager”, a count of the values of Proposal Id’s listed as “Projects

Approved”, and count of proposal ID’s minus Unique Proposal ID’s

divided by count of total proposal ids times 100 and listed as

“PercentProjectsWithinHrs”. All of these will values will come from the

Project Managers Table

JOIN: Table People with Table Project Managers based on People ID and

Manager ID respectively

LEFT JOIN: Join Table Project Managers and Table proposals but

return all rows from table Project Managers and only matching

values from table people based on Manager ID

and People ID respectively. Join Table Project Managers and

Project Manager 2 Manager ID and Approving manager respectively.

To calculate the values selected from PM 2 for adding in the table we

select the sum of actual hours as “Total Hours” from table proposal

progress, Approving Manager from proposals, and Proposal Id from

proposals, join table proposal progress with table people on proposal

ID, Group everything by Approving manager, proposed hours, and

proposal ID and only select those with Projects having a sum of

actual hours greater than its proposed hours.

GROUP BY: Group these values based on first and last name from the project

managers table and Total hours and approving manager from the

Project Managers 2 table.

ORDER BY: Order the table by percent of projects within hours for each manger

from highest (best performing managers) to lowest.

**BUSINESS QUESTION 9:** Ref BR5. Who were the last three employees to work on a proposal

that was above its proposed hours?

SHOW: Show the top 3 employees concatenated names as ‘Employee’,

which are formed by pulling the first name, an inserted space, and

last name from table people, also concat the employees manager

name by pulling first name, an inserted space, and last name from

table people as ‘Manager’. Also show proposal name from table

proposals, actual hours from proposal progress, and current date as

‘Time Entry Date’ from table time entries.

JOIN: Join table proposal progress with table time entries based on time

entry ID. Join table time entries with table employees based on

employee ID. Join table employees with the table project team based

on employee ID. Join the table project team with the table project

manager based on manager ID. Join table people with the table

project manager based on people ID, which is the same as manager

ID. Join table people with table employees based on people ID,

which is the same as employee ID. Join table people with table

proposals based on proposal ID.

SUBQUERY 1: Show proposal ID from table proposals. Join table proposals with

table proposal progress based on proposal ID. Table proposals’

proposal ID equals table proposal progress’s proposal ID. Group

results by proposal ID then proposed hours, having the sum of actual

hours from table proposal progress be greater than proposed hours

from table proposals.

ORGANIZE: Where proposal ID from table people equals SUBQUERY 1. Order

results by ‘Time Entry Date’ with the results descending from newest

to oldest.

**BUSINESS QUESTION 10:** Ref BR4. Which employee(s) have worked on projects that were

above the proposed hours and how many hours did they work on

the project?

SHOW: Proposal Name listed as “ProposalBeyondProposedHours), a First

and last name combined together listed as “EmployeeName”, the

sum of the actual hours worked listed as “Hoursworked” from the

Proposal Progress table. From the proposal progress table we will

show Total Hours listed as “ProjectTotalHours and Proposed Hours

listed as “ProjectProposedHours”

JOIN: Table time Entries with Table Proposal Progress on Time Entry ID all

from the proposal progress table

LEFT JOIN: Join Table Proposal Projects and Table Employees but

return all rows from table Proposal Projects and only matching

values from table Employees based on Employee ID. Join Table

Proposal Projects and Table Employees but return all rows from

table Proposal Projects and only matching values from table

Employees based on Employee ID.

JOIN: To calculate the values selected from PR for adding in the table

We select the sum of actual hours as “Total Hours” from table

proposal progress, Proposal Name, and Proposal Id from

proposals, join table proposal progress with table people on

Proposal ID, Group everything by Proposal ID, proposed hours,

and proposal name and only select those with Employees

having a sum of actual hours greater than their proposed hours.

Join the table based on Proposal ID.

GROUP BY: The Proposal ID, Employee First and Last Name, Proposal

Name, Total Hours, and Proposed Hours