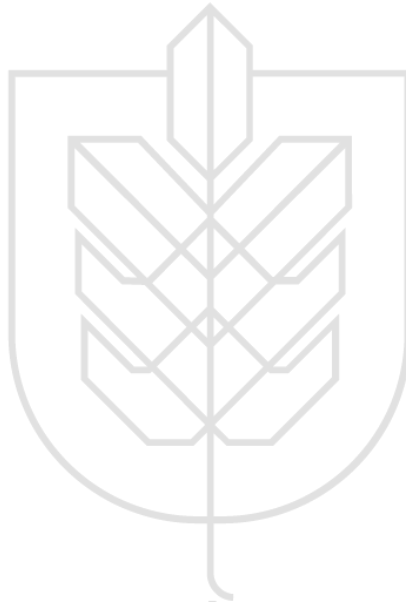


DATA MANAGEMENT & SQL

DAT-5305

A2: Advertising Management Platform



Boston, Master's in Business Analytics
14th of March 2021

Prof. Maithily Erande

Group 20:
Jack Daoud
Max Lembke

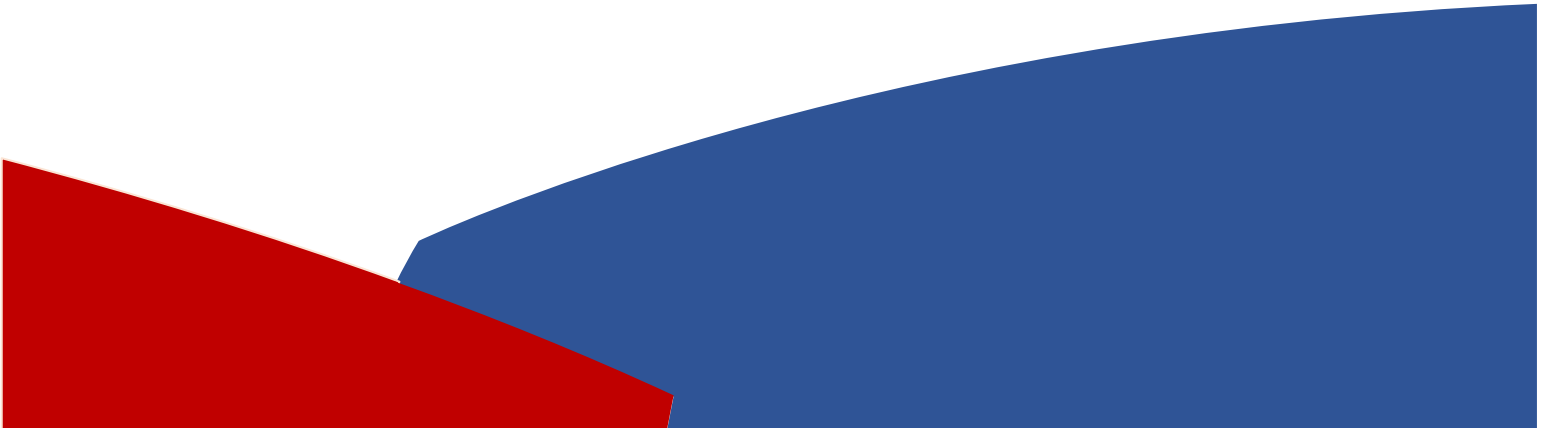


Table of Contents

Executive Summary	1
Reports	2
Report 1: Social media (SM) platform performance	2
Report 2: Social media (SM) performance campaign.....	2
Report 3: Unidentified late campaigns	2
Report 4: Late Campaigns and their current costs	2
Report 5: Sunk costs cancelled campaigns	3
Report 6: Highest cost overrun employee.....	3
Report 7: Running ads per employee.....	3
Report 8: Total ads per product for product manager.....	3
System Flow	4
i. New Campaign	4
ii. New Ad for a campaign.....	4
iii. Data updates (Employee changes &Metrics)	5
iv. Accessing/Querying Data	5
System Entities	6
Entity-Relational (ER) Model.....	8
Stored Procedures.....	9
<i>Procedure 01 – Update Budget.....</i>	<i>9</i>
Test 01.....	10
Test 02.....	10
<i>Procedure 02 - Update Status & Status tracking.....</i>	<i>11</i>
Test 01.....	12
Test 02.....	13

Test 03.....	14
SQL Queries	15
<i>Report 01 – Social Media (SM) Platform Performance</i>	<i>15</i>
<i>Report 02 – SM Platform Performance per Campaign</i>	<i>16</i>
<i>Report 02.1 – Stored Procedure Method (Better UX than 02 – for a specific campaign).....</i>	<i>17</i>
<i>Report 03 – Unidentified Late Campaigns</i>	<i>18</i>
<i>Report 04 – Costs of Late Campaigns</i>	<i>18</i>
<i>Report 05 - Sunk Cost of Cancelled Campaigns.....</i>	<i>19</i>
<i>Report 06 - Highest Overrun Cost Employee</i>	<i>20</i>
<i>Report 07 – Running Ads per Employee</i>	<i>20</i>
<i>Report 08 – Total Ads for Product for Product Manager</i>	<i>21</i>
<i>Report 09 – Marketing Budget & Cost for Product Manager</i>	<i>22</i>
Appendix.....	23
<i>Create Table Script.....</i>	<i>23</i>
<i>Import Data Script</i>	<i>30</i>
<i>References</i>	<i>36</i>

Executive Summary

The marketing and advertising space has undoubtedly undergone fundamental and rapid change with the rise of social media and other digital tools. Though most companies have become deeply engulfed in this digital space, managing a plethora of platforms and posts for each campaign underway has proved a herculean task for many. Unarguably, the lack of tools enabling a convenient summarization of insights from these platforms and absent link to internal systems, has led to many companies to adopt “hodge-podge” solutions for advertisement management. These unstable and unscalable solutions are often based upon a mix of excel, google sheets and other third-party platforms, making them prone to errors and difficult to use.

Proposed Solution:

The solution presented in this consultancy shall bridge the gap in the current market, providing a scalable and coherent approach to advertising management, combining the tracking of campaigns, their related ads on different platforms and internal operations. The advertising management platform (AMP) shall therefore streamline and centralize the processes of planning, executing, monitoring, optimizing, and reporting on campaigns and their associated ads (Thompson, 2019).

Scope & Limitations:

The scope of the proposed solution has been limited to a use-case for medium to small companies, that operate a centralized marketing team, in which product or service owners' interface with the marketing team. The marketing team is thereby the primary user of the tool designed and the client. Furthermore, the scope is limited by currently only supporting the upload of Facebook and LinkedIn data. This is due to the authors only having access to company pages on these platforms and being able to use their Application programming interface (API) to download reports. The solution presented in this paper can easily be expanded by creating another table for additional social media platforms, that mirror the download of data from those additional social media platforms.

Reports

Report 1: Social media (SM) platform performance

The objective of this report is to identify which social media platform is reaching the largest amount of people. It should enable the client to see the total across all campaigns and ads, thereby being very high level. Finally, a metric should be utilized that is comparable across all platforms (in this case, total impressions).

Report 2: Social media (SM) performance campaign

Each campaign has a differing number of posts and budget associated to boosting its performance on a given social media platform. In order to evaluate the effectiveness, of each campaign the amount of impressions reached needs to be presented and the cost per impression needs be displayed. This will allow for a more differentiated analysis.

Report 3: Unidentified late campaigns

Humans make mistakes and similarly to outdated google sheets, people are prone to not updating records such as a campaigns launch running late. The database user, especially individuals in management roles needs to be able check if all campaigns that are running late have been identified as such. This output can and should then be used to update the record accordingly.

Report 4: Late Campaigns and their current costs

The fact that a campaign runs late is not always an issue, especially if the campaign was delayed and the with it the use of funds. This can become an issue however if a campaign is dragging out and then exceeding its budget as a boosted post, for example, is still accumulating costs for the company. The department needs to be able to have an overview of delayed campaigns and the current status of expenditure.

Report 5: Sunk costs cancelled campaigns

Campaigns are often designed or launched before a product or service is actually shipping or even ready to be shipped. Issues in manufacturing or logistics can result in the associated campaigns and thereby ads being cancelled even though outward facing advertisement has started. An overview of the financial impact of cancelled campaigns needs to be generated to adjust forecasting and other financial planning.

Report 6: Highest cost overrun employee

Executives and managers need to be able to evaluate employee performance not only using metric performance. Arguably if an employee continuously overspends on campaigns this is not in the interest of the company. The employees with the biggest cost overrun over the budget needs to be identified before employee evaluations.

Report 7: Running ads per employee

In order to evaluate the current human resource (HR) utilization to potentially accept or reject the running of new campaigns/ads, managers need to be able to view the current (HR) utilization. An output of the number of ads per employee needs to be provided.

Report 8: Total ads per product for product manager

Product owners, that the marketing department interfaces with, often would like to receive an overview of how many ads are currently running across the product/services they are responsible for. The marketing department needs to be able to provide such information quickly without looking these up manually on each channel.

Report 9: Marketing budget and cost for product manager

Product owners would also like to know if they are under or overspending. In addition to the amount of ads, the marketing team needs to be able to generate reports that highlight the budgeted and used budget for every products campaign if the product owner requests this.

System Flow

The system, as presented in this version, is scalable and could interface with APIs directly. Several entries of data, that are described as manual processes below, can be automated when linking these to other databases. This would minimize the manual entry to only an update of new campaigns and ads.

i. New Campaign

The marketing team receives a request from a product or service owner in the company to create a marketing campaign for their product or service. The contact point, if not recorded already will be recorded and a new campaign will be added for the product/service. After being in contact with the respective product owner, details such as budget, marketing employee responsible and time frame will be recorded through a manual upload into the system. This initial recording shall remain a manual procedure.

ii. New Ad for a campaign

After the marketing team has instantiated a campaign, they will decide to run certain ads. These will be manually recorded in the respective table by the employee responsible for the campaign. This will include marketing related information such as the platform advertised on, cost and other descriptors. This recording shall also remain a manual procedure.

iii. Data updates (Employee changes & Metrics)

Employees and their current team are manual uploads in the current database. Any changes to this in the database, as of right now would need to be changed or added manually. A more mature company could retrieve this information from their human resource department via an API access from SAP (for example).

Any updates to the metrics can only be achieved by downloading the respective summary report from the platform and uploading this to the database. However, no changes are necessary in this, as the column names chosen match those of the platform csv downloads. Similarly, this can easily be automated by using the API access provided by platforms to pull this information on a daily basis and update accordingly. The presented solution would allow for an immediate link to the API of each platform, as it follows the column name structure of the csv downloads and developer API documentation. The client would only be required to deploy the database and follow the instructions provided in the developer documentation. No changes are needed in the structure of the database.

iv. Accessing/Querying Data

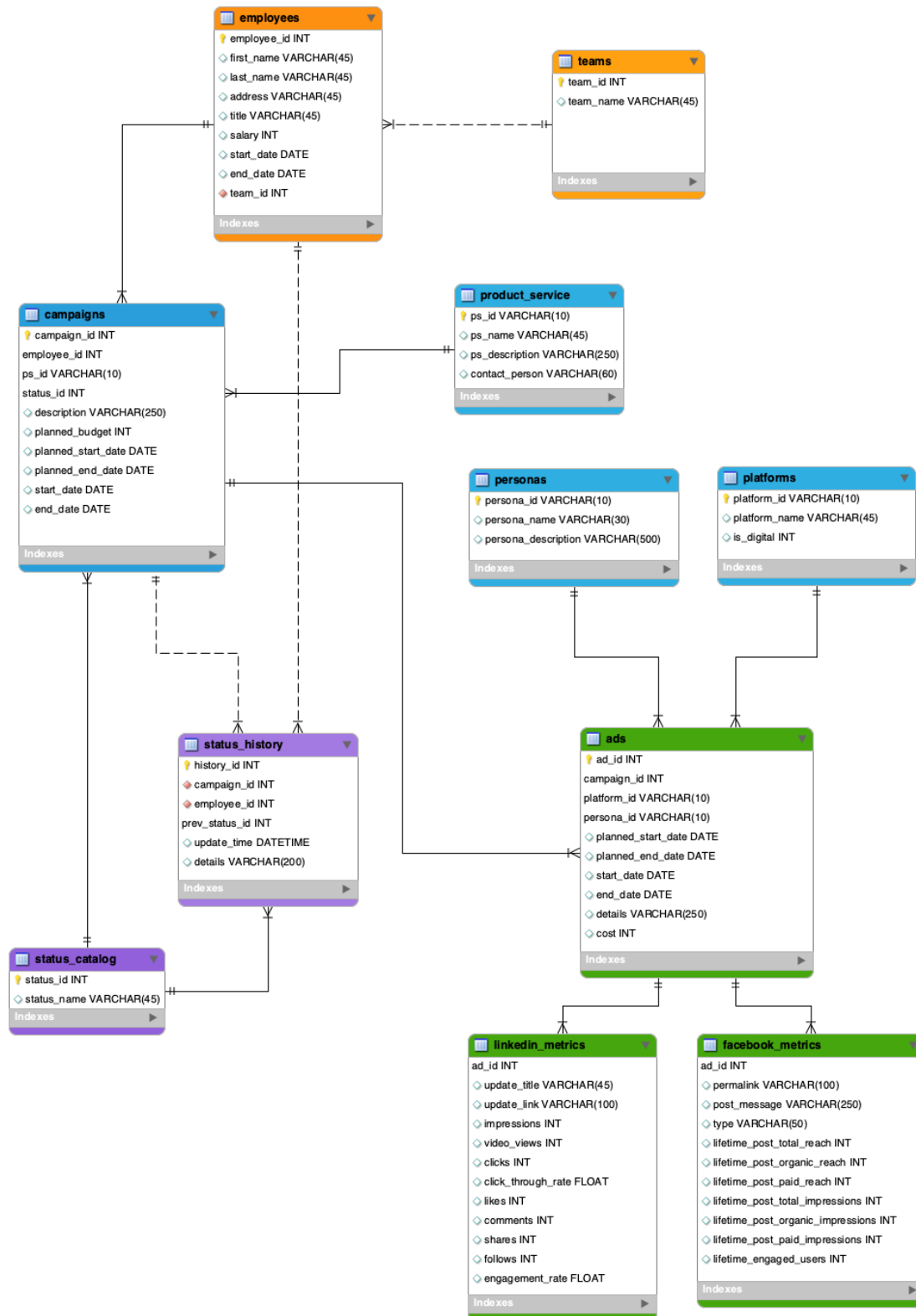
The database presented, does not contain any personal identifiable information of customers or target of advertisements. Hence access to query the information related to campaigns, can be provided to all members of the marketing team, if the marketing team pleases to do so. Provided that the client utilizing the system does not wish to share information related to costs and budgets, the employee responsible for each campaign can run queries and share the outputs with his team. Changes and modifications should only be made by system admins. Status or update changes, that do require updating on a more frequent basis have been tackled through procedures below.

System Entities

Table Name	Primary Key(s)	Foreign Key(s)	Description
employees	employee_id	team_id	Employee master table contains all necessary information about the employees within the marketing department.
teams	team_id		Team master table contains the team names & team IDs and is linked only to the employee table.
campaigns	campaign_id	employee_id, ps_id, status_id	Campaigns table contains all the core information for marketing campaigns. It is considered both a child and master table. Child because it inherits fields from other tables (such as employees, product/service, and status catalog), and master because it's the parent of the Ads table.
product_service	ps_id		Product service master table contains the information about the company's products. This includes name, description and contact person of the respective product.
personas	persona_id		Persona master table contains information on persona names and their very long and detailed descriptions. This table is a parent to Ads.
platforms	platform_id		Platform master table contains information on platforms/channels and whether or not they are digital. This table is a parent to Ads.

status_history	history_id	campaign_id, employee_id	This child's table is meant to keep track of campaign status changes overtime. It includes links to the campaign and the employee conducting the status update as well as an update timestamp and details for future reference.
status_catalog	status_id		This master table is a small catalog of status types (their IDs and names) and it's linked to both Campaigns and Status History
ads	ad_id	campaign_id, platform_id, persona_id	Ads table is considered both a master and child table. It's a child of Campaigns, Personas, and Platforms in order to have a scope of the ad in terms of target, channel, product, etc. It's a master table of social media platform metrics.
linkedin_metrics		ad_id	This is a pure child table derived from Ads. It contains metrics related to LinkedIn ads.
facebook_metrics		ad_id	This is a pure child table derived from Ads. It contains metrics related to Facebook ads.

Entity-Relational (ER) Model



Stored Procedures

Procedure 01 – Update Budget

Objective: Update the budget of a campaign

Key result: The ability to update the overall budget of a campaign without resorting to the help of an external system or database admin

Input: Campaign ID & Updated Budget

Output: No visual output, the related campaigns budget will be updated

```
CREATE DEFINER='root'@'localhost' PROCEDURE `update_budget`(  
  IN in_camp_id INT,  
  IN in_budget INT  
)  
BEGIN  
  DECLARE done INT DEFAULT FALSE;  
  DECLARE cur_camp_id INT;  
  DECLARE cur1 CURSOR FOR  
    SELECT campaign_id  
    FROM campaigns;  
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
  OPEN cur1;  
read_loop: LOOP  
    FETCH cur1 into cur_camp_id;  
  IF done THEN  
    LEAVE read_loop;  
  END IF;  
  
  UPDATE campaigns  
  SET planned_budget = planned_budget + in_budget  
    WHERE campaign_id = cur_camp_id  
    AND campaign_id = in_camp_id;  
END LOOP;  
CLOSE cur1;  
END
```

Test 01

```
SELECT
campaign_id AS "Campaign ID",
employee_id AS "Employee ID",
    planned_budget AS "Budget"
FROM campaigns
WHERE    campaign_id    =    123;
```

	Campaign ID	Employee ID	Budget	
►	123	2	1000	

```
CALL update_budget(123, 1000);
```

	Campaign ID	Employee ID	Budget	
►	123	2	2000	

Test 02

```
SELECT
campaign_id AS "Campaign ID",
    employee_id AS "Employee ID",
    planned_budget AS "Budget"
FROM campaigns
WHERE    campaign_id    =    524;
```

	Campaign ID	Employee ID	Budget	
►	524	2	19000	

```
CALL update_budget(524, 6000);
```

	Campaign ID	Employee ID	Budget	
►	524	2	25000	

Test 03

```
SELECT
campaign_id AS "Campaign ID",
    employee_id AS "Employee ID",
    planned_budget AS "Budget"
FROM campaigns
WHERE    campaign_id    =    219;
```

	Campaign ID	Employee ID	Budget	
►	219	7	3000	

```
CALL update_budget(219, -1000);
```

	Campaign ID	Employee ID	Budget	
►	219	7	2000	

Procedure 02 - Update Status & Status tracking

Objective: Update the status of a campaign

Key result: The ability to update the status of the campaign and record the changes including who made the change to status and why in a history table for archiving

Input: Employee ID (who is making the change), Campaign ID, New Status & Description (reason for change/update of status)

Output: No visual output, the related new status of the campaign shall be updated in the campaigns table and the old status, time of change, employee ID and reason for change should be recorded in the history table

```
CREATE DEFINER='root'@'localhost' PROCEDURE `update_status`(IN in_emp INT, IN
in_campaign_id INT, IN new_status INT, IN details_in VARCHAR(200))
BEGIN
```

```
DECLARE old_status INT;
DECLARE temp_cid INT;
DECLARE cur_time DATETIME;
```

```
SELECT c.campaign_id, c.status_id
INTO temp_cid, old_status
FROM campaigns AS c
WHERE c.campaign_id = in_campaign_id;
```

```
SELECT NOW()
INTO cur_time;
```

```
UPDATE campaigns as c2
SET c2.status_id = new_status
WHERE c2.campaign_id = temp_cid;
```

```
INSERT INTO `ad_mgmt`.`status_history`
(`campaign_id`,`employee_id`,`prev_status_id`,`update_time`,`details`)
VALUES (temp_cid, in_emp, old_status, cur_time, details_in);
```

```
END
```

Test 01

```
SELECT
c.campaign_id AS "Campaign ID",
c.status_id AS "Status ID",
sc.status_name AS "Status",
c.planned_start_date AS "Planned Start Date",
c.planned_end_date AS "Planned End Date",
c.start_date AS "Actual Start Date",
c.end_date AS "End Date",
CURDATE() AS "Today's Date"
FROM campaigns AS c, status_catalog AS sc
WHERE campaign_id = 524
AND c.status_id = sc.status_id;
```

	Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶	524	1	In Process	2021-01-20	2021-02-20	2021-01-25	NULL	2021-03-14

CALL update_status(2,524,3,"The campaign is running late as identified by query, updating database to reflect this");

	Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶	524	3	Overtime	2021-01-20	2021-02-20	2021-01-25	NULL	2021-03-14

Inserted record in status_history:

ID	Campaign ID	Employee ID	Previous Status ID	Update Timestamp	Details
1	456	10	1	2021-02-27 00:51:21	Change from in progress to hold
2	986	10	1	2021-02-10 10:23:44	Placed on hold due to human resource constraints
▶ 3	986	5	2	2021-02-12 12:55:45	Placed on in progress after being delayed
4	986	3	1	2021-03-02 07:10:20	Changed to complete, as it is done
5	219	3	1	2021-02-27 15:03:06	Campaign was cancelled due to product issue
6	94	9	1	2021-03-08 10:05:12	Change to on hold as focus for campaign ran i...
7	498	9	1	2021-01-23 09:10:22	Campaign was cancelled due to logistical issues
8	157	8	1	2021-03-10 15:27:01	Campaign is on-going past the planned end date
9	524	2	1	2021-03-14 17:29:23	The campaign is running late as identified by qu...

Test 02

```
SELECT
```

```
c.campaign_id AS "Campaign ID",
c.status_id AS "Status ID",
sc.status_name AS "Status",
c.planned_start_date AS "Planned Start Date",
c.planned_end_date AS "Planned End Date",
c.start_date AS "Actual Start Date",
c.end_date AS "End Date",
CURDATE() AS "Today's Date"
FROM campaigns AS c, status_catalog AS sc
WHERE campaign_id = 424
AND c.status_id = sc.status_id;
```

	Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶	456	2	On Hold	2021-04-20	2021-05-20	NULL	NULL	2021-03-14

```
CALL update_status(4,456,1,"Change from hold to in progress as there is enough HR capacity now");
```

	Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶	456	1	In Progress	2021-04-20	2021-05-20	NULL	NULL	2021-03-14

Inserted record in status_history:

ID	Campaign ID	Employee ID	Previous Status ID	Update Timestamp	Details
▶ 1	456	10	1	2021-02-27 00:51:21	Change from in progress to hold
2	986	10	1	2021-02-10 10:23:44	Placed on hold due to human resource constraints
3	986	5	2	2021-02-12 12:55:45	Placed on in progress after being delayed
4	986	3	1	2021-03-02 07:10:20	Changed to complete, as it is done
5	219	3	1	2021-02-27 15:03:06	Campaign was cancelled due to product issue
6	94	9	1	2021-03-08 10:05:12	Change to on hold as focus for campaign ran into issues
7	498	9	1	2021-01-23 09:10:22	Campaign was cancelled due to logistical issues
8	157	8	1	2021-03-10 15:27:01	Campaign is on-going past the planned end date
9	524	2	1	2021-03-14 17:29:23	The campaign is running late as identified by query, update...
10	456	4	2	2021-03-14 17:35:55	Change from hold to in progress as there is enough HR ca...

Test 03

```
SELECT
```

```
c.campaign_id AS "Campaign ID",
c.status_id AS "Status ID",
sc.status_name AS "Status",
c.planned_start_date AS "Planned Start Date",
c.planned_end_date AS "Planned End Date",
c.start_date AS "Actual Start Date",
c.end_date AS "End Date",
CURDATE() AS "Today's Date"
FROM campaigns AS c, status_catalog AS sc
WHERE campaign_id = 424
AND c.status_id = sc.status_id;
```

Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶ 456	1	In Progress	2021-04-20	2021-05-20	NULL	NULL	2021-03-14

```
CALL update_status(4,456,1,"Change from hold to in progress as there is enough HR capacity now");
```

Campaign ID	Status ID	Status	Planned Start Date	Planned End Date	Actual Start Date	End Date	Today's Date
▶ 456	2	On Hold	2021-04-20	2021-05-20	NULL	NULL	2021-03-14

Inserted record in status_history:

ID	Campaign ID	Employee ID	Previous Status ID	Update Timestamp	Details
▶ 1	456	10	1	2021-02-27 00:51:21	Change from in progress to hold
2	986	10	1	2021-02-10 10:23:44	Placed on hold due to human resource constraints
3	986	5	2	2021-02-12 12:55:45	Placed on in progress after being delayed
4	986	3	1	2021-03-02 07:10:20	Changed to complete, as it is done
5	219	3	1	2021-02-27 15:03:06	Campaign was cancelled due to product issue
6	94	9	1	2021-03-08 10:05:12	Change to on hold as focus for campaign ran into issues
7	498	9	1	2021-01-23 09:10:22	Campaign was cancelled due to logistical issues
8	157	8	1	2021-03-10 15:27:01	Campaign is on-going past the planned end date
9	524	2	1	2021-03-14 17:29:23	The campaign is running late as identified by query, update...
10	456	4	2	2021-03-14 17:35:55	Change from hold to in progress as there is enough HR ca...
11	456	4	1	2021-03-14 17:37:43	Error in setting to in progress before, its actually still on hold

SQL Queries

Report 01 – Social Media (SM) Platform Performance

```
CREATE VIEW lm_imp_total AS
```

```
(SELECT sum(impressions) AS "LinkedIn_Total"
```

```
FROM linkedin_metrics);
```

```
CREATE VIEW fb_imp_total AS
```

```
(SELECT sum(lifetime_post_total_impressions) AS "Facebook_Total"
```

```
FROM facebook_metrics);
```

```
SELECT LinkedIn_Total AS " LinkedIn Total Impressions", Facebook_Total AS " Facebook  
Total Impressions"
```

```
FROM lm_imp_total
```

```
CROSS JOIN fb_imp_total;
```

	LinkedIn Total Impressions	Facebook Total Impressions	
▶	191137	179773	

Report 02 – SM Platform Performance per Campaign

```
CREATE VIEW lm_imp AS
(SELECT ad_id, sum(impressions) AS
"imp_lm"
FROM linkedin_metrics
GROUP BY ad_id);
```

```
CREATE VIEW fb_imp AS
(SELECT ad_id,
sum(lifetime_post_total_impressions) AS
"imp_f"
FROM facebook_metrics
GROUP BY ad_id);
```

```
CREATE VIEW campaign_overview_stats
AS (SELECT a.campaign_id,
SUM((SELECT SUM(imp_f)
FROM fb_imp
WHERE fb_imp.ad_id = a.ad_id))
AS "Total_Imp_Facebook",
SUM((SELECT SUM(imp_lm)
FROM lm_imp
WHERE lm_imp.ad_id = a.ad_id))
AS "Total_Imp_LinkedIn"
FROM ads AS a
GROUP BY a.campaign_id);
```

```
CREATE VIEW FB_cost AS (
SELECT a.campaign_id, sum(a.cost) as
"Facebook_Cost"
```

```
FROM ads AS a
GROUP BY a.campaign_id, a.platform_id
HAVING platform_id IN ("FB"));
```

```
CREATE VIEW LI_cost AS (
SELECT a.campaign_id, sum(a.cost) AS
"LinkedIn_Cost"
FROM ads AS a
GROUP BY a.campaign_id, a.platform_id
HAVING platform_id IN ("LI"));
```

```
SELECT
cos.campaign_id AS "Campaign ID",
cos.Total_Imp_Facebook AS "Total
Impressions (FB)",
cos.Total_Imp_LinkedIn AS "Total
Impressions (LI)",
Facebook_Cost AS "Total Cost (FB)",
LinkedIn_Cost AS "Total Cost (LI)",
Facebook_Cost/cos.Total_Imp_Facebook
AS "Cost per Impression (FB)",
LinkedIn_Cost/cos.Total_Imp_LinkedIn AS
"Cost per Impression (LI)"
FROM campaign_overview_stats AS cos,
FB_cost, LI_cost
WHERE cos.campaign_id =
FB_cost.campaign_id
AND cos.campaign_id =
LI_cost.campaign_id;
```

Campaign ID	Total Impressions (FB)	Total Impressions (LI)	Total Cost (FB)	Total Cost (LI)	Cost per Impression (FB)	Cost per Impression (LI)
▶ 123	22430	7471	350	200	0.0156	0.0268
157	2560	39802	1800	458	0.7031	0.0115
219	13517	46529	700	900	0.0518	0.0193
498	3057	69940	100	290	0.0327	0.0041
524	103975	13911	1000	478	0.0096	0.0344
852	12475	8524	754	650	0.0604	0.0763
986	21759	4960	400	600	0.0184	0.1210

Report 02.1 – Stored Procedure Method (Better UX than 02 – for a specific campaign)

```
CREATE DEFINER='root'@'localhost'
PROCEDURE
`overview_socials_campaigns`(IN
in_campaign_id INT)
BEGIN
```

```
CREATE VIEW lm_imp AS
(SELECT ad_id, sum(impressions) AS
"imp_lm"
FROM linkedin_metrics
GROUP BY ad_id);
```

```
CREATE VIEW fb_imp AS
(SELECT ad_id,
sum(lifetime_post_total_impressions) AS
"imp_f"
FROM facebook_metrics
GROUP BY ad_id);
```

```
CREATE VIEW campaign_overview_stats
AS (SELECT a.campaign_id,
SUM((SELECT SUM(imp_f)
FROM fb_imp
WHERE fb_imp.ad_id = a.ad_id))
AS "Total_Imp_Facebook",
SUM((SELECT SUM(imp_lm)
FROM lm_imp
WHERE lm_imp.ad_id = a.ad_id))
AS "Total_Imp_LinkedIn"
FROM ads AS a
GROUP BY a.campaign_id);
```

```
CREATE VIEW FB_cost AS (
SELECT a.campaign_id,sum(a.cost) as
"Facebook_Cost"
FROM ads AS a
```

```
GROUP BY a.campaign_id, a.platform_id
HAVING platform_id IN ("FB"));
```

```
CREATE VIEW LI_cost AS (
SELECT a.campaign_id,sum(a.cost) AS
"LinkedIn_Cost"
FROM ads AS a
GROUP BY a.campaign_id, a.platform_id
HAVING platform_id IN ("LI"));
```

```
SELECT
cos.campaign_id AS "Campaign ID",
cos.Total_Imp_Facebook AS "Total
Impressions (FB)",
cos.Total_Imp_LinkedIn AS "Total
Impressions (LI)",
Facebook_Cost AS "Total Cost (FB)",
LinkedIn_Cost AS "Total Cost (LI)",
Facebook_Cost/cos.Total_Imp_Facebook
AS "Cost per Impression (FB)",
LinkedIn_Cost/cos.Total_Imp_LinkedIn AS
"Cost per Impression (LI)"
FROM campaign_overview_stats AS cos,
FB_cost, LI_cost
WHERE cos.campaign_id =
FB_cost.campaign_id
AND cos.campaign_id =
LI_cost.campaign_id;
```

```
DROP VIEW lm_imp;
DROP VIEW fb_imp;
DROP VIEW campaign_overview_stats;
DROP VIEW li_cost;
DROP VIEW fb_cost;
END
```

Report 03 – Unidentified Late Campaigns

```

SELECT
campaign_id AS "Campaign ID",
    status_id AS "Status ID",
    planned_end_date AS "Planned End Date",
    end_date AS "Actual End Date",
    CURDATE() AS "Current Date"
FROM campaigns
WHERE planned_end_date < CURDATE()
AND end_date IS NULL
AND status_id != 3;

```

	Campaign ID	Status ID	Planned End Date	Actual End Date	Current Date
▶	524	1	2021-02-20	NULL	2021-03-14

Report 04 – Costs of Late Campaigns

```

SELECT
c.campaign_id AS "Campaign ID",
    c.planned_budget AS "Planned Budget",
    SUM(a.cost) AS "Actual Cost",
    c.planned_budget-SUM(a.cost) AS "Cost Delta"
FROM campaigns AS c, ads AS a
WHERE c.campaign_id = a.campaign_id
AND c.status_id = 3
GROUP BY c.campaign_id;

```

	Campaign ID	Planned Budget	Actual Cost	Cost Delta
▶	157	3000	3637	-637

Report 05 - Sunk Cost of Cancelled Campaigns

```
SELECT
    c.campaign_id AS "Campaign ID",
    COUNT(a.ad_id) AS "Number of Ads",
    c.planned_budget AS "Planned Budget",
    SUM(a.cost) AS "Sunk Cost"
FROM campaigns AS c, ads AS a
WHERE c.campaign_id = a.campaign_id
AND c.status_id = 4
GROUP BY c.campaign_id;
```

	Campaign ID	Number of Ads	Planned Budget	Sunk Cost	
▶	219	4	3000	1600	
	498	3	1000	889	

Report 06 - Highest Overrun Cost Employee

```

SELECT
    CONCAT(e.last_name, ' ', e.first_name) AS "Employee",
    c.campaign_id AS "Campaign ID",
    CASE
        WHEN SUM(a.cost)-c.planned_budget > 0
        THEN SUM(a.cost)-c.planned_budget
        ELSE 0
    END AS "Highest Overrun Cost"
FROM employees AS e, campaigns AS c, ads AS a
WHERE e.employee_id = c.employee_id
AND c.campaign_id = a.campaign_id
AND c.status_id = 3
GROUP BY c.campaign_id;

```

Employee	Campaign ID	Highest Overrun Cost
▶ Voldemort, Lord	157	637

Report 07 – Running Ads per Employee

```

SELECT CONCAT(e.last_name, ' ', e.first_name) AS "Employee",
    COUNT(a.ad_id) AS "Number of Ads"
FROM employees AS e, ads AS a, campaigns AS c
WHERE a.campaign_id = c.campaign_id
AND c.employee_id = e.employee_id
AND c.status_id IN (1, 3)
GROUP BY CONCAT(e.last_name, ' ', e.first_name)
ORDER BY COUNT(a.ad_id) DESC;

```

Employee	Number of Ads
▶ Granger, Hermione	6
Potter, Harry	5
Chang, Cho	4
Voldemort, Lord	4

Report 08 – Total Ads for Product for Product Manager

```
SELECT
    ps.contact_person AS "Product/Service Owner",
    ps.ps_name AS "Product/Service",
    COUNT(a.ad_id) AS "Count of Ads"
FROM campaigns AS c, ads AS a, product_service AS ps
WHERE c.campaign_id = a.campaign_id
AND c.ps_id = ps.ps_id
AND ps.contact_person = "Thomas, Nussman"
AND a.planned_end_date > CURDATE()
GROUP BY a.campaign_id;
```

	Product/Service Owner	Product/Service	Count of Ads	
▶	Thomas, Nussman	service_new	4	

Report 09 – Marketing Budget & Cost for Product Manager

```

CREATE VIEW cmp_ps AS
(SELECT
    c.ps_id,
    SUM(c.planned_budget) AS "Marketing_Budget",
    SUM(a.cost) AS "Total_Expense"
FROM campaigns AS c, ads AS a
WHERE c.campaign_id = a.campaign_id
GROUP BY c.ps_id);

SELECT
    ps.contact_person AS "Product/Service Owner",
    ps.ps_name AS "Product/Service",
    cp.Marketing_Budget AS "Marketing Budget",
    cp.Total_Expense AS "Total Expense"
FROM product_service AS ps, cmp_ps AS cp
WHERE ps.ps_id = cp.ps_id
AND ps.contact_person = "Jack, Black";

```

	Product/Service Owner	Product/Service	Marketing Budget	Total Expense	
►	Jack, Black	cool_thing	5000	1050	
	Jack, Black	original_product	15000	2489	

Appendix

Create Table Script

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
-- Schema ad_mgmt
-- -----
```

```
-- -----
-- Schema ad_mgmt
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `ad_mgmt` DEFAULT CHARACTER SET utf8 ;
USE `ad_mgmt` ;
```

```
-- -----
-- Table `ad_mgmt`.`teams`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`teams` (
  `team_id` INT NOT NULL,
  `team_name` VARCHAR(45) NULL,
  PRIMARY KEY (`team_id`))
ENGINE = InnoDB;
```

```
-- -----
-- Table `ad_mgmt`.`employees`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`employees` (
  `employee_id` INT NOT NULL,
```

```

`first_name` VARCHAR(45) NULL,
`last_name` VARCHAR(45) NULL,
`address` VARCHAR(45) NULL,
`title` VARCHAR(45) NULL,
`salary` INT NULL,
`start_date` DATE NULL,
`end_date` DATE NULL,
`team_id` INT NOT NULL,
PRIMARY KEY (`employee_id`),
INDEX `fk_employees_teams1_idx` (`team_id` ASC) VISIBLE,
CONSTRAINT `fk_employees_teams1`
  FOREIGN KEY (`team_id`)
    REFERENCES `ad_mgmt`.`teams` (`team_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ad_mgmt`.`personas`
-----
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`personas` (
  `persona_id` VARCHAR(10) NOT NULL,
  `persona_name` VARCHAR(30) NULL,
  `persona_description` VARCHAR(500) NULL,
  PRIMARY KEY (`persona_id`))
ENGINE = InnoDB;

```

```

-----
-- Table `ad_mgmt`.`product_service`
-----
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`product_service` (
  `ps_id` VARCHAR(10) NOT NULL,
  `ps_name` VARCHAR(45) NULL,
  `ps_description` VARCHAR(250) NULL,
  `contact_person` VARCHAR(60) NULL,
  PRIMARY KEY (`ps_id`))
ENGINE = InnoDB;

```

```
-- -----  
-- Table `ad_mgmt`.`status_catalog`  
-- -----  
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`status_catalog` (  
  `status_id` INT NOT NULL,  
  `status_name` VARCHAR(45) NULL,  
  PRIMARY KEY (`status_id`))  
ENGINE = InnoDB;  
  
-- -----  
-- Table `ad_mgmt`.`campaigns`  
-- -----  
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`campaigns` (  
  `campaign_id` INT NOT NULL,  
  `employee_id` INT NOT NULL,  
  `ps_id` VARCHAR(10) NOT NULL,  
  `status_id` INT NOT NULL,  
  `description` VARCHAR(250) NULL,  
  `planned_budget` INT NULL,  
  `planned_start_date` DATE NULL,  
  `planned_end_date` DATE NULL,  
  `start_date` DATE NULL,  
  `end_date` DATE NULL,  
  PRIMARY KEY (`campaign_id`, `employee_id`, `ps_id`, `status_id`),  
  INDEX `fk_campaigns_employees1_idx` (`employee_id` ASC) VISIBLE,  
  INDEX `fk_campaigns_product_service2_idx` (`ps_id` ASC) VISIBLE,  
  INDEX `fk_campaigns_status_catalog1_idx` (`status_id` ASC) VISIBLE,  
  CONSTRAINT `fk_campaigns_employees1`  
    FOREIGN KEY (`employee_id`)  
    REFERENCES `ad_mgmt`.`employees` (`employee_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_campaigns_product_service2`  
    FOREIGN KEY (`ps_id`)  
    REFERENCES `ad_mgmt`.`product_service` (`ps_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_campaigns_status_catalog1`
```

```
FOREIGN KEY (`status_id`)
REFERENCES `ad_mgmt`.`status_catalog` (`status_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `ad_mgmt`.`platforms`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`platforms` (
  `platform_id` VARCHAR(10) NOT NULL,
  `platform_name` VARCHAR(45) NULL,
  `is_digital` INT NULL,
  PRIMARY KEY (`platform_id`))
ENGINE = InnoDB;
```

```
-- -----
-- Table `ad_mgmt`.`ads`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`ads` (
  `ad_id` INT NOT NULL,
  `campaign_id` INT NOT NULL,
  `platform_id` VARCHAR(10) NOT NULL,
  `persona_id` VARCHAR(10) NOT NULL,
  `planned_start_date` DATE NULL,
  `planned_end_date` DATE NULL,
  `start_date` DATE NULL,
  `end_date` DATE NULL,
  `details` VARCHAR(250) NULL,
  `cost` INT NULL,
  PRIMARY KEY (`ad_id`, `campaign_id`, `platform_id`, `persona_id`),
  INDEX `fk_ads_campaigns1_idx` (`campaign_id` ASC) VISIBLE,
  INDEX `fk_ads_platforms1_idx` (`platform_id` ASC) VISIBLE,
  INDEX `fk_ads_personas1_idx` (`persona_id` ASC) VISIBLE,
  CONSTRAINT `fk_ads_campaigns1`
    FOREIGN KEY (`campaign_id`)
    REFERENCES `ad_mgmt`.`campaigns` (`campaign_id`)
    ON DELETE NO ACTION
```

```

    ON UPDATE NO ACTION,
CONSTRAINT `fk_ads_platforms1`
    FOREIGN KEY (`platform_id`)
    REFERENCES `ad_mgmt`.`platforms` (`platform_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_ads_personas1`
    FOREIGN KEY (`persona_id`)
    REFERENCES `ad_mgmt`.`personas` (`persona_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ad_mgmt`.`linkedin_metrics`
-----

```

```

CREATE TABLE IF NOT EXISTS `ad_mgmt`.`linkedin_metrics` (
  `ad_id` INT NOT NULL,
  `update_title` VARCHAR(45) NULL,
  `update_link` VARCHAR(100) NULL,
  `impressions` INT NULL,
  `video_views` INT NULL,
  `clicks` INT NULL,
  `click_through_rate` FLOAT NULL,
  `likes` INT NULL,
  `comments` INT NULL,
  `shares` INT NULL,
  `follows` INT NULL,
  `engagement_rate` FLOAT NULL,
  PRIMARY KEY (`ad_id`),
  INDEX `fk_linkedin_metrics_ads1_idx` (`ad_id` ASC) VISIBLE,
  CONSTRAINT `fk_linkedin_metrics_ads1`
    FOREIGN KEY (`ad_id`)
    REFERENCES `ad_mgmt`.`ads` (`ad_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```
-----  
-- Table `ad_mgmt`.`facebook_metrics`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`facebook_metrics` (  
  `ad_id` INT NOT NULL,  
  `permalink` VARCHAR(100) NULL,  
  `post_message` VARCHAR(250) NULL,  
  `type` VARCHAR(50) NULL,  
  `lifetime_post_total_reach` INT NULL,  
  `lifetime_post_organic_reach` INT NULL,  
  `lifetime_post_paid_reach` INT NULL,  
  `lifetime_post_total_impressions` INT NULL,  
  `lifetime_post_organic_impressions` INT NULL,  
  `lifetime_post_paid_impressions` INT NULL,  
  `lifetime_engaged_users` INT NULL,  
  PRIMARY KEY (`ad_id`),  
  INDEX `fk_facebook_metrics_ads1_idx` (`ad_id` ASC) VISIBLE,  
  CONSTRAINT `fk_facebook_metrics_ads1`  
    FOREIGN KEY (`ad_id`)  
      REFERENCES `ad_mgmt`.`ads` (`ad_id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `ad_mgmt`.`status_history`  
-----
```

```
CREATE TABLE IF NOT EXISTS `ad_mgmt`.`status_history` (  
  `history_id` INT NOT NULL AUTO_INCREMENT,  
  `campaign_id` INT NOT NULL,  
  `employee_id` INT NOT NULL,  
  `prev_status_id` INT NOT NULL,  
  `update_time` DATETIME NULL,  
  `details` VARCHAR(200) NULL,  
  PRIMARY KEY (`history_id`, `prev_status_id`),  
  INDEX `fk_status_history_status_catalog1_idx` (`prev_status_id` ASC) VISIBLE,  
  INDEX `fk_status_history_campaigns1_idx` (`campaign_id` ASC) VISIBLE,  
  INDEX `fk_status_history_employees1_idx` (`employee_id` ASC) VISIBLE,  
  CONSTRAINT `fk_status_history_status_catalog1`
```

```
FOREIGN KEY (`prev_status_id`)
REFERENCES `ad_mgmt`.`status_catalog` (`status_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_status_history_campaigns1`
FOREIGN KEY (`campaign_id`)
REFERENCES `ad_mgmt`.`campaigns` (`campaign_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_status_history_employees1`
FOREIGN KEY (`employee_id`)
REFERENCES `ad_mgmt`.`employees` (`employee_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```


Import Data Script

-- Platforms Table

```
INSERT INTO platforms
(platform_id, platform_name, is_digital)
VALUES
('FB','Facebook',1),
('LI','LinkedIn',1),
('NP','Newspaper',0),
('RD','Radio',0),
('TV','Television',0);
```

-- Teams Table

```
INSERT INTO teams
(team_id, team_name)
VALUES
(1,'Content'),
(2,'Social Media'),
(3,'Paid Media');
```

-- Employees Table

```
INSERT INTO employees
(employee_id, first_name, last_name, address, title, salary, start_date, end_date, team_id)
VALUES
(1,'Hermione','Granger','The Burrow','Charms Tutor',100000,'2018-02-14',NULL,1),
(2,'Harry','Potter','Hogwarts','Lord',400000,'2018-02-14',NULL,2),
(3,'Ron','Weasley','Little Hangleton','Quidditch Captin',50000,'2021-02-14',NULL,3),
(4,'Draco','Malfoy','Malfoy Manor','Snake',250000,'2020-02-14',NULL,1),
(5,'Lord','Voldemort','Little Whinging','Dark Lord',400000,'2015-05-23',NULL,2),
(6,'Bellatrix','Lestrangle','Godric's Hollow','Senior Warlock',100000,'2017-06-08',NULL,3),
(7,'Sirius','Black','Spinner's End','Godparent',300000,'2015-06-09','2021-03-01',1),
(8,'Ginny','Weasley','Shell Cottage','Deputy Headmaster',50000,'2015-09-10',NULL,2),
(9,'Lucius','Malfoy','Malfoy Manor','Director of Magical Secuirity',100000,'2016-12-25',NULL,3),
(10,'Cho','Chang','Hogsmeade Station','Master of Death',125000,'2019-04-12',NULL,1);
```

-- Personas Table

```
INSERT INTO personas
(persona_id, persona_name, persona_description)
VALUES
('AA','Attentive Andy','Interested in occupation related content');
```

```
("BC","Bored Charlie","Interested in funny content"),
("MM","Mother Mandy","Interested in familial content"),
("BB","Busy Bob","Interested in news");
```

```
-- Product / Service Table
```

```
INSERT INTO product_service
(ps_id, ps_name, ps_description, contact_person)
VALUES
("P1","cool_thing","A cool product that we are selling and we really want to push it. ","Jack, Black"),
("S1","service_helpful ","We have this new service people will think is helpful. ","Thomas, Nussman"),
("P2","helpful_solution ","This solution will solve our customers efficiency constraints.","Sarah, Tod"),
("P3","original_product","Our original product we became famous for. ","Jack, Black"),
("S2","service_new","A brand new service we have in our portfolio. ","Thomas, Nussman");
```

```
-- Status Catalog Table
```

```
INSERT INTO status_catalog
(status_id, status_name)
VALUES
(1,"In Process"),
(2,"On Hold"),
(3,"Overtime"),
(4,"Cancelled"),
(5,"Complete");
```

```
-- Campaign Table
```

```
INSERT INTO campaigns
(campaign_id,employee_id,description,status_id,planned_budget,
planned_start_date,planned_end_date,start_date,end_date,ps_id)
VALUES
(123,2,"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut metus diam. Morbi sollicitudin elit sit amet sapien blandit vestibulum.",1,1000,"2021-02-27","2021-03-27","2021-02-27",NULL,"P1"),
(456,6,"Sed imperdiet euismod libero, nec tincidunt leo placerat quis. Vestibulum interdum magna nulla, non semper turpis porta at. Cras ultricies nulla ut enim pulvinar, ut luctus quam tincidunt.",2,1500,"2021-04-20","2021-05-20",NULL,NULL,"P1"),
(986,1,"Duis varius dui enim. Nullam vulputate gravida ligula, vitae mattis erat ornare quis.",5,1000,"2021-02-14","2021-03-02","2021-02-14","2021-03-02","S1"),
```

(524,2,"Suspendisse ut condimentum risus. Fusce magna neque, feugiat at ligula tincidunt, elementum suscipit odio. ",1,19000,"2021-01-20","2021-02-20","2021-01-25",NULL,"P2"),
 (219,7,"Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed mollis eu est at posuere.",4,3000,"2021-02-11","2021-03-13","2021-02-11","2021-02-28","P3"),
 (94,8,"Fusce accumsan tortor nunc, ac porttitor odio molestie non. Pellentesque vehicula risus sapien, sed iaculis nunc imperdiet ut",2,2500,"2021-03-02","2021-04-10",NULL,NULL,"S2"),
 (498,4,"Aenean ullamcorper, metus sed ultrices consectetur, orci purus rhoncus magna, nec semper purus mauris sit amet purus.",4,1000,"2021-01-10","2021-03-10","2021-01-12","2021-01-23","P3"),
 (157,5,"Nunc auctor nec magna id tempus. Cras tempor iaculis lectus, id cursus nisl cursus eu. Integer sed lorem tincidunt, egestas enim eget, dapibus est.",3,3000,"2021-02-23","2021-03-10","2021-02-24",NULL,"S1"),
 (852,10,"Donec vitae nisl ac orci luctus tincidunt non quis ante. Donec aliquam nulla mauris, ut sagittis nisl venenatis vel. Sed sodales sem vel purus eleifend, ut sollicitudin massa pretium.",1,17500,"2021-02-10","2021-03-15","2021-02-10",NULL,"S2");

-- Status History Table

INSERT INTO status_history

(history_id,campaign_id,employee_id,prev_status_id,update_time,details)

VALUES

(1,456,10,1,"2021-02-27 00:51:21","Change from in progress to hold "),
 (2,986,10,1,"2021-02-10 10:23:44","Placed on hold due to human resource constraints"),
 (3,986,5,2,"2021-02-12 12:55:45","Placed on in progress after being delayed "),
 (4,986,3,1,"2021-03-02 07:10:20","Changed to complete, as it is done "),
 (5,219,3,1,"2021-02-27 15:03:06","Campaign was cancelled due to product issue "),
 (6,94,9,1,"2021-03-08 10:05:12","Change to on hold as focus for campaign ran into issues "),
 (7,498,9,1,"2021-01-23 09:10:22","Campaign was cancelled due to logistical issues"),
 (8,157,8,1,"2021-03-10 15:27:01","Campaign is on-going past the planned end date");

-- Ads Table

INSERT INTO ads

(ad_id,persona_id,platform_id,campaign_id,planned_start_date,

planned_end_date,start_date,end_date,details,cost)

VALUES

(9854,"AA","FB",123,"2021-02-27","2021-03-13","2021-02-27","2021-03-13","Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut metus diam. Morbi sollicitudin elit sit amet sapien blandit vestibulum.",200),

(9738,"BB","FB",524,"2021-01-20","2021-02-03","2021-01-25","2021-02-08","Sed imperdiet euismod libero, nec tincidunt leo placerat quis. Vestibulum interdum magna nulla, non semper turpis porta at. Cras ultricies nulla ut enim pulvinar, ut luctus quam tincidunt.",100),
(9123,"MM","FB",986,"2021-02-14","2021-03-02","2021-02-14","2021-03-02","Duis varius dui enim. Nullam vulputate gravida ligula, vitae mattis erat ornare quis. ",400),
(9548,"BB","FB",524,"2021-02-08","2021-02-20","2021-03-06",NULL,"Suspendisse ut condimentum risus. Fusce magna neque, feugiat at ligula tincidunt, elementum suscipit odio. ",300),
(9328,"BB","FB",219,"2021-02-11","2021-02-25","2021-02-11","2021-02-25","Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed mollis eu est at posuere.",600),
(8743,"AA","FB",219,"2021-02-25","2021-03-11","2021-02-25","2021-02-28","Fusce accumsan tortor nunc, ac porttitor odio molestie non. Pellentesque vehicula risus sapien, sed iaculis nunc imperdiet ut",100),
(9874,"BC","FB",498,"2021-01-10","2021-03-10","2021-01-12","2021-01-23","Aenean ullamcorper, metus sed ultrices consectetur, orci purus rhoncus magna, nec semper purus mauris sit amet purus.",100),
(7802,"MM","FB",157,"2021-02-23","2021-03-10","2021-02-24",NULL,"Nunc auctor nec magna id tempus. Cras tempor iaculis lectus, id cursus nisl cursus eu. Integer sed lorem tincidunt, egestas enim eget, dapibus est.",1800),
(7809,"BB","FB",852,"2021-02-10","2021-03-15","2021-02-10",NULL,"Donec vitae nisl ac orci luctus tincidunt non quis ante. Donec aliquam nulla mauris, ut sagittis nisl venenatis vel. Sed sodales sem vel purus eleifend, ut sollicitudin massa pretium.",754),
(9076,"BC","FB",123,"2021-03-11","2021-03-25","2021-03-11",NULL,"Duis vitae vehicula velit. Suspendisse pellentesque, dolor vel hendrerit tincidunt, nibh risus iaculis eros, a ultrices nunc erat vitae turpis.",150),
(8793,"MM","FB",524,"2021-01-20","2021-02-03","2021-01-25","2021-02-08","Quisque euismod pretium felis, et finibus metus. Etiam hendrerit lobortis libero ac ornare.",600),
(2123,"BB","LI",123,"2021-02-27","2021-03-13","2021-02-27","2021-03-13","Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut metus diam. Morbi sollicitudin elit sit amet sapien blandit vestibulum.",100),
(2126,"AA","LI",852,"2021-02-10","2021-03-15","2021-02-28",NULL,"Sed imperdiet euismod libero, nec tincidunt leo placerat quis. Vestibulum interdum magna nulla, non semper turpis porta at. Cras ultricies nulla ut enim pulvinar, ut luctus quam tincidunt.",200),
(2128,"BC","LI",986,"2021-02-14","2021-03-02","2021-02-14","2021-03-02","Duis varius dui enim. Nullam vulputate gravida ligula, vitae mattis erat ornare quis. ",600),
(2134,"AA","LI",524,"2021-02-08","2021-02-20","2021-03-06",NULL,"Suspendisse ut condimentum risus. Fusce magna neque, feugiat at ligula tincidunt, elementum suscipit odio. ",478),

(1312,"BC","LI",219,"2021-02-11","2021-02-25","2021-02-11","2021-02-25","Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed mollis eu est at posuere.",400),

(3212,"AA","LI",219,"2021-02-25","2021-03-11","2021-02-25","2021-02-28","Fusce accumsan tortor nunc, ac porttitor odio molestie non. Pellentesque vehicula risus sapien, sed iaculis nunc imperdiet ut",500),

(3123,"BC","LI",157,"2021-02-23","2021-02-28","2021-02-24","2021-03-01","Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed mollis eu est at posuere.",102),

(3578,"AA","LI",157,"2021-02-28","2021-03-10","2021-03-01",NULL,"Fusce accumsan tortor nunc, ac porttitor odio molestie non. Pellentesque vehicula risus sapien, sed iaculis nunc imperdiet ut",356),

(3420,"BC","LI",852,"2021-02-10","2021-03-15","2021-02-10","2021-02-28","Aenean ullamcorper, metus sed ultrices consectetur, orci purus rhoncus magna, nec semper purus mauris sit amet purus.",450),

(2315,"AA","LI",123,"2021-03-11","2021-03-25","2021-03-11",NULL,"Nunc auctor nec magna id tempus. Cras tempor iaculis lectus, id cursus nisl cursus eu. Integer sed lorem tincidunt, egestas enim eget, dapibus est.",100),

(2594,"BC","LI",498,"2021-01-10","2021-03-10","2021-01-12","2021-01-23","Donec vitae nisl ac orci luctus tincidunt non quis ante. Donec aliquam nulla mauris, ut sagittis nisl venenatis vel. Sed sodales sem vel purus eleifend, ut sollicitudin massa pretium.",290),

(6534,"MM","NP",498,"2021-01-10","2021-03-10","2021-01-12","2021-01-23","Duis vitae vehicula velit. Suspendisse pellentesque, dolor vel hendrerit tincidunt, nibh risus iaculis eros, a ultrices nunc erat vitae turpis.",499),

(6578,"AA","NP",157,"2021-02-23","2021-03-01","2021-02-24",NULL,"Quisque euismod pretium felis, et finibus metus. Etiam hendrerit lobortis libero ac ornare.",1379),

(4576,"BC","RD",123,"2021-03-04","2021-03-11","2021-03-04","2021-03-11","Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut metus diam. Morbi sollicitudin elit sit amet sapien blandit vestibulum.",500),

(5747,"AA","RD",524,"2021-01-30","2021-02-06","2021-01-30","2021-02-06","Sed imperdiet euismod libero, nec tincidunt leo placerat quis. Vestibulum interdum magna nulla, non semper turpis porta at. Cras ultricies nulla ut enim pulvinar, ut luctus quam tincidunt.",900),

(7689,"BC","TV",852,"2021-02-10","2021-03-15","2021-02-10",NULL,"Duis varius dui enim. Nullam vulputate gravida ligula, vitae mattis erat ornare quis. ",10000),

(7896,"MM","TV",524,"2021-01-30","2021-02-06","2021-01-30","2021-02-06","Suspendisse ut condimentum risus. Fusce magna neque, feugiat at ligula tincidunt, elementum suscipit odio.",10000);

-- Facebook Metrics

INSERT INTO facebook_metrics

```
(ad_id,permalink,post_message,type,lifetime_post_total_reach,lifetime_post_organic_reach,
lifetime_post_paid_reach,lifetime_post_total_impressions,lifetime_post_organic_impressions,
lifetime_post_paid_impressions,lifetime_engaged_users)
```

```
VALUES
```

```
(9854,"facebook.com/ourpage/posts/this","We have this awesome thing for you ","Photo /
Video",12263,6622,5641,9933,4172,5761,11098),
(9738,"facebook.com/ourpage/posts/that","This is a cool product","Photo
Carousel",76526,55099,21427,34437,24450,9987,55481),
(9123,"facebook.com/ourpage/posts/product","Join us
","Slideshow",31999,20799,11200,21759,11532,10227,26879),
(9548,"facebook.com/ourpage/posts/fun","Do this now ","Photo /
Video",31237,25927,5310,14057,1827,12229,22647),
(9328,"facebook.com/ourpage/posts/hi","Check this out ","Photo
Album",14853,9209,5644,12922,6849,6073,13888),
(8743,"facebook.com/ourpage/posts/cool","We love our
customers","Slideshow",2833,227,2606,595,30,565,1714),
(9874,"facebook.com/ourpage/posts/thing","Big
update","Canvas",8046,1931,6115,3057,397,2660,5552),
(7802,"facebook.com/ourpage/posts/peeps","Breaking news","Photo
Album",3084,648,2436,2560,205,2355,2822),
(7809,"facebook.com/ourpage/posts/join","Amazing stuff","Photo
Album",23537,4237,19300,12475,9356,3119,18006),
(9076,"facebook.com/ourpage/posts/ad","What are you wating for? ","Photo
Carousel",34714,19440,15274,12497,3499,8998,23606),
(8793,"facebook.com/ourpage/posts/party","This is fun
","Canvas",57793,41033,16760,55481,34398,21083,56637);
```

```
-- LinkedIn Metrics
```

```
INSERT INTO linkedin_metrics
```

```
(ad_id,update_title,update_link,impressions,video_views,clicks,
click_through_rate,likes,comments,shares,follows,engagement_rate)
```

```
VALUES
```

```
(2123,"We have this awesome thing for you
","ourwebsite.com/this",1740,1352,654,19.74,506,411,87,139,23.22),
(2126,"This is a cool
product","ourwebsite.com/that",3364,NULL,2024,73.09,2794,1016,1028,265,81.21),
(2128,"Join us ","ourwebsite.com/product",4960,NULL,1539,29.88,4438,1750,1930,334,63.57),
(2134,"Do this now
","ourwebsite.com/fun",13911,NULL,5998,62.4,4664,3605,1935,1112,65.69),
```

(1312,"Check this out
 ","ourwebsite.com/hi",43295,12862,30917,33.56,30701,29842,3569,2681,62.16),
 (3212,"We love our
 customers","ourwebsite.com/cool",3234,745,209,0.25,2366,2168,1827,299,1.3),
 (3123,"Big update","ourwebsite.com/thing",12728,NULL,9613,0.62,204,203,49,1183,10.35),
 (3578,"Breaking
 news","ourwebsite.com/peeps",27074,NULL,16553,1.83,14263,6629,9898,615,30.46),
 (3420,"Amazing stuff","ourwebsite.com/join",5160,NULL,2464,3.56,1046,159,136,263,8.08),
 (2315,"What are you wating for?
 ","ourwebsite.com/ad",5731,NULL,3520,7.23,5537,201,1142,193,48.21),
 (2594,"This is fun
 ","ourwebsite.com/party",69940,10345,43305,63.93,49962,40275,803,4591,77.97);

References

Thompson, E. (2019, August 13). *What is an Advertising Management Platform*. Lumenad.

<https://lumenad.com/what-is-an-advertising-management-platform/>