Data Management & Sql

ADVERTISING MANAGEMENT MODEL PROPOSAL

Tommaso Mazzucco/Natalia Gomez

Hult International Business School

- PART 1 – Executive overview of the system:

Outsmart labs it's a digital marketing agency that is growing at a fast-paced by acquiring various international clients. They currently manage all the digital marketing initiatives of multiple clients trying to grow their brands' digital presence.

As many other advertising agencies, Outsmart Labs struggles to centralize all their different campaigns and get unique consumer insights because of the increase of multiple devices and channels. These omnichannel experiences are making for agencies increasingly complex to gain unique insights about the clients leads, their consumer journey, and to understand what campaigns are driving results and which ones are not, and how to optimize accordingly. Therefore, they need an advertising management database capable of providing them a holistic unified view of all the leads and campaign results per account and monitoring their interactions with the campaigns across all channels.

Furthermore, Outsmart Labs' organizational data is highly fragmented at several levels, leading to siloed data and duplicate account records in different databases. This creates a data quality challenge for Outsmart Labs, like storing monthly invoices to their respective account and registering the expenditures per account per month.

Based on the above, the purpose of the following system would be to create an efficient advertising management platform capable of integrating all available data about each account and respective campaign into one view, so Outsmart Labs could deliver a compelling customer experience across all channels. By solving all the data daze and fragmentation, the advertising management model will centralize the processes and technology required to run cross-channel campaigns.

- PART 2 – FLOW OF THE SYSTEM -

This advertising management database system will fundamentally focus on campaign data. For each campaign, the idea is to store all related accounts and leads, what product was being advertised, what employees were in charge, the overall performance of the campaign, and all the campaign results store in one platform. We'll also manage invoices between the agency and its clients.

The idea is that the model is capable of storing details related to the services we provide to clients. We can expect that the client will ask for an all-in-one solution for his needs. For example, the client could ask us to run Google Ads and at the same time manage paid social campaigns.

I. ACCOUNTS

The first section of the model will revolve around their account management platform. Thus, the flow of the system starts with the accounts table. Each account has a unique identifier, so it is easier to track all the movement across the client's database with that unique ID. This account table is created based on the need of storing your client's data and contact details in one platform.

II. INVOICES

The invoices table will allow you to manage each monthly account bill more seamlessly and store it in your system. This table also allows you to send any details about that specifically invoice to the clients and is connected to each client's unique identifier.

III. PAYMENTS

The idea behind the payments table is to allow the company to monitor each transaction, by using each invoice Identifier. Invoice ID and Payments ID show as the relation the account table has to these ID's.

IV. AGENCY EMPLOYEES

The agency has few employees at the moment, but having them in a system is essential, and more so if you will keep growing at the pace you have been in the last few years. It is also important to know which employee was doing what for what campaign and advertising effort. To this category, the first one is Employees, with all their identifying details and their respective Job Title.

V. PRODUCTS

The model continues with our Products table. We'll advertise a specific product (or service) or multiple products for each account for each campaign. A link to each product will be provided, in order to store the product's location at the moment it is available. This attribute is specific to the product table and is not present in the campaigns table.

VI. CAMPAIGNS

The entire system revolves around the agency campaigns. Without these, none of the other tables could exist. For each campaign, the idea is to store all related accounts and leads, what product was being advertised, what employees were in charge, the overall performance of the campaign, and all the campaign results store in one platform. In the campaign table, we'll manage the list of all campaigns our agency has done for a client and its respective channel type. Each campaign is strictly related to only one client and product. But we could have multiple campaigns for the same client at the same time or consecutively. Hence, we must be able to build a solid and concise data infrastructure that can facilitate us to see the results of each present and past campaign.

VII. CAMPIGN_RESULTS

We need to get a holistic picture of how each campaign is performing on a monthly and Quartey basis and get a unified view of the results per account. This is going to be very beneficial for the agency when preparing the monthly performance reports for each account. We decided to take four critical conversions metrics that will help us identify if a campaign performs or didn't perform:

- Total_reach
- Total_engagement
- Total_impressions
- Total_conversions
- ROI
- Clicks
- CTR

VIII. LEADS

We need to capture a bigger scope of data related to each lead per account. The way to promote the clients' products will be directly related to the information you gather from them. We understood this need and based on it and on the logic of the business, so the leads table accompanies campaigns. All following tables are directly or indirectly related to their information. Therefore the amount of data collected from them will be crucial. The relation between this table could potentially allow Outsmart Labs to understand the average age of the customers per account, their gender, when they buy, why, or the frequency of their purchases. We included the channel type and the campaign_id to actively track which channel is bringing the most customers and potentially strengthen your efforts to that specific channel. With this amount of information, the possibility of focusing our advertising efforts on a client-centric approach is higher (e.g., email marketing).

- PART 3 – DATABASE STRUCTURE -

Account

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Account_ID	INT	NO	PRIMARY	Unique Identifier of the Account
Account_name	VARCHAR (45)	NO		Name of the client
Account_address	VARCHAR (45)	NO		Address of the client
Account email	VARCHAR (45)	NO		Email of client

Products

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Product_ID	INT	NO	PRIMARY	Unique Identifier for products
Campaign_ID	VARCHAR (45)	NO		Unique Identifier of campaign
Product_name	VARCHAR (45)	NO		Name of product
Product_link	VARCHAR (45)	NO		Link that redirects to product
Account_ID	VARCHAR (45)	NO		Unique Identifier of the Account

Agency Employees

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Employee_ID	INT	NO	PRIMARY	Unique Identifier Agency Employees
Employee_name	VARCHAR (45)	NO		Name of Employee of agency
Employee_last_name	VARCHAR (45)	NO		Last name of Employee
JobTitle	VARCHAR (45)	NO		Role in the Agency

Invoices

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Invoice_ID	INT	NO	PRIMARY	Unique Identifier for invoices
Invoice_amount	INT	NO		Amount $ corresponding to invoice
Other_details	LONGTEXT	YES		Details relatively to clients
Account_ID	INT	NO		Unique identifier for Accounts
Invoice_issued	DATE			Date when the invoice is schedule

STATUS	VARCHAR (45)			The status of the invoice to see if it was paid or unpaid

Campaigns

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Campaign_ID	INT	NO	Primary	Unique Identifier Campaign
Campaign_name	VARCHAR (45)	NO		Name of Campaign
Status	TINYTEXT	NO		Status of campaign
Start_date	DATE	NO		Start date of Campaign
End_date	DATE	NO		End date of campaign
Description	LONGTEXT	YES		Description of Campaign
Channel_type	VARCHAR (45)	NO		Type of channel
Budget	INT	NO		Amount of Budget
Product_ID	INT	NO		Unique Identifier Product
Employee_ID	INT	NO		Unique Identifier Employee
Objective	VARCHAR (45)	NO		The campaign objective
Creative_type	VARCHAR (45)	NO		The format of the campaign(e.g.video, display)

Payments

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Payment_ID	INT	NO	PRIMARY	Unique Identifier Campaign
Payment_Amount	INT	NO		Amount of Payment
Other_details	INT			Details of Payments
Invoice_ID	INT	NO		Unique Identifier for Invoices

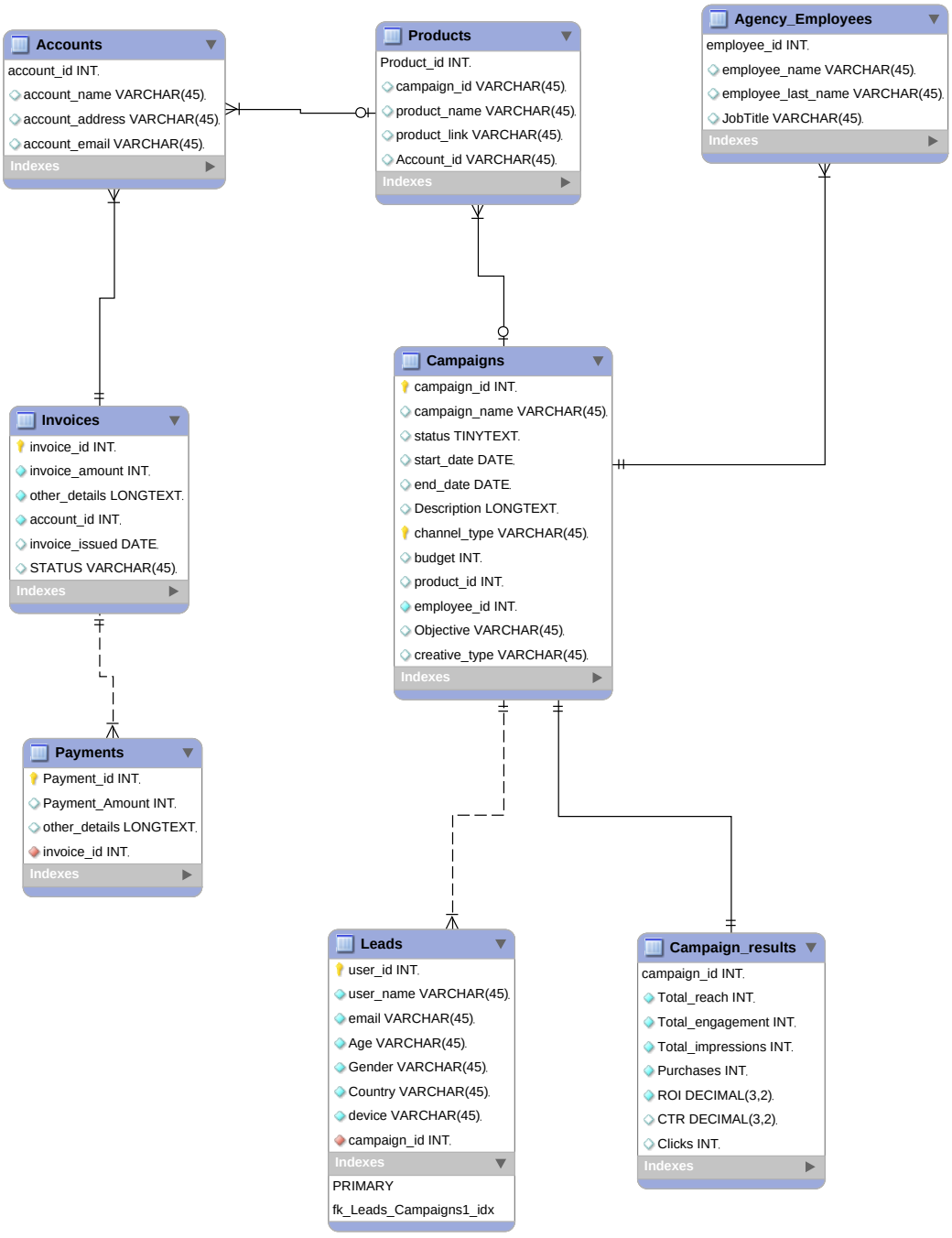
Leads

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Leads_ID	INT	NO	PRIMARY	Unique Identifier Leads
User_name	VARCHAR (45)	NO		User name
Email	VARCHAR (45)	NO		Email of User
Age	VARCHAR (45)	NO		User's age
Gender	VARCHAR (45)	NO		User's gender
Country	VARCHAR (45)	NO		Country where user Is from
Device	VARCHAR (45)	NO		User's device used
Campaign_ID	INT	NO		Unique Identifier of Campaign

Campaign_results

FIELD	DATATYPE	NULL	KEY	DESCRIPTION
Total_reach	INT	NO		# people who have seen campaign
Total_Engagement	INT	NO		# reactions and interactions to campaings
Purchases	INT	NO		# of purchases
Campaign_ID	INT	NO		Unique Identifier of Campaign
Total_Impressions	int	no		Total # of exposures to the campaign
ROI	Decimal (3,2)	NO		Return on Investment per each campaign
CTR	Decimal(3,2)	NO		# of clicks received on the campaigns per # of impressions
Clicks	INT	NO		# of people who click on the campaign

- PART 4 – ENTITY-RELATIONAL MODEL –



- PART 5 – SQL QUERIES TO BUILD THE REPORT –

I. This example shows how to determine which campaigns are reaching a given demographic. This would be crucial in order to optimize campaigns accordingly to the audiences.

```
SELECT gender AS "User Gender", country AS "User Country", age AS "Customer Age", campaign_id,
        COUNT(user_id) AS "Number of Customers"
FROM leads
GROUP BY country, gender, Age, campaign_id
ORDER BY gender, country ASC;
```

User Gender	User Country	Customer Age	campaign_id	Number of Custom...
F	Brazil	31	6	1
F	China	22	5	1
F	Colombia	25	1	1
F	Peru	20	7	1
F	Spain	25	9	1
F	USA	17	10	1
F	USA	33	3	1
M	Germany	39	8	1
M	Italy	24	2	1
M	USA	18	4	1

II. Show the most recent Invoice per account

```
select a.account_id, b.invoice_id, a.account_name, b.invoice_amount,b.invoice_issued, b.STATUS
from accounts a, invoices b
where b.invoice_issued = (select max(b1.invoice_issued)
                from invoices b1
                where b.invoice_id = b1.invoice_id)
and a.account_id = b.Account_id;
```

account_id	invoice_id	account_name	invoice_amount	invoice_issued	STATUS
1009	1078	BoingToys	25000	2021-12-01	OVERDUE
1450	1368	DesignDistrict	5000	2021-03-01	UNPAID
1360	1645	CoyoTaco	15000	2021-03-03	PAID
1580	2078	MoxyHotel	60000	2021-03-01	PAID
6040	2348	Serena	12900	2021-03-01	OVERDUE
1890	2575	TravelAir	10000	2021-03-01	PAID
2020	2681	HiltlonCaribbean	8000	2021-03-01	OVERDUE
5060	4366	NeuroSpa	30000	2021-03-01	UNPAID
1220	4368	ToySmart	5000	2021-03-01	OVERDUE
1340	9076	SilviaTcherassi	1000	2021-03-01	PAID

III. Get a holistic unified view of how the most recent campaigns are performing

```
SELECT distinct  a.campaign_id, b.total_reach, b.total_engagement, b.total_impressions,
b.purchases, b.CTR ,c.start_date,c.end_date
from leads a, Campaign_results b, campaigns c
where c.end_date = (select max(c1.end_date)
                                                    from campaigns c1
                                                                where  c1.campaign_id
= c.campaign_id)

and a.campaign_id = b.campaign_id
and b.campaign_id = c.campaign_id
order by b.CTR desc;
```

campaign_id	total_reach	total_engagement	total_impressio...	purchases	CTR	start_date	end_date
6	75000	46000	700	9	8.57	2021-02-02	2022-09-09
8	164000	145000	180000	1000	0.83	2021-09-09	2021-06-01
10	612	311	642	3	0.66	2021-03-01	2021-04-14
9	1825	1200	1927	7	0.46	2021-09-09	2021-04-12
1	1390	1000	2000	50	0.40	2021-03-01	2021-04-04
4	20000	5000	50000	600	0.34	2020-07-09	2021-08-09
3	500	160	800	100	0.12	2021-01-01	2021-05-08
5	300	130	600	2	0.08	2020-09-10	2021-08-10
2	8000	2400	10000	185	0.06	2021-03-01	2021-04-04
7	1000	67	1500	1	0.03	2021-01-02	2021-05-04

IV. Only show the campaigns that are active per account and who is the campaign lead

```
SELECT c.status, c.campaign_id, a.account_id, a.account_name, c.employee_id,
concat_ws("," ,d.employee_name, d.employee_last_name) AS "Employee Name"
from campaigns c, products p, accounts a, agency_employees d
where status = "Active"
And c.product_id = p.product_id
And p.account_id = a.account_id
and c.employee_id = d.employee_id;
```

status	campaign_id	account_id	account_name	employee_id	Employee Name
ACTIVE	1	1009	BoingToys	1998	Tommaso,Mazzucco
ACTIVE	3	1340	SilviaTcherassi	1990	Mariana,Ballesteros
ACTIVE	4	5060	NeuroSpa	6321	martha,Aycardi
ACTIVE	6	2020	HiltlonCaribbean	1111	Marc,Friedman
ACTIVE	7	1890	TravelAir	1693	Mariana,Gutierrez
ACTIVE	8	6040	Serena	1990	Mariana,Ballesteros
ACTIVE	9	1360	CoyoTaco	3476	Alexandra,Rodriguez
ACTIVE	11	1580	MoxyHotel	1998	Tommaso,Mazzucco
ACTIVE	12	1360	CoyoTaco	3476	Alexandra,Rodriguez

V. This example shows how to analyze audiences. Learn which audiences impressions are reaching, and determine if some audiences perform better than others. This knowledge can help balance unique cookie count (putting ads in front of a lot of users) and quality (narrow targeting and viewable impressions), depending on the campaign objectives.

```
    SELECT a.campaign_id, a.device,b.total_impressions,b.total_reach,b.clicks,
    COUNT(DISTINCT a.user_id) AS uniques,
    ROUND(COUNT(*) / COUNT(DISTINCT a.user_id), 1) AS frequency
    FROM leads a, campaign_results b
    where a.campaign_id =b.campaign_id
GROUP BY a.campaign_id, b.clicks,b.total_impressions,b.total_reach, a.device;
```

	campaign_id	device	total_impressio...	total_reach	clicks	uniques	frequency
▶	1	mobile	2000	1390	800	1	1.0
	2	desktop	10000	8000	650	1	1.0
	3	mobile	800	500	100	1	1.0
	4	desktop	50000	20000	17000	1	1.0
	5	mobile	600	300	50	1	1.0
	6	mobile	700	75000	60000	1	1.0
	7	mobile	1500	1000	37	1	1.0
	8	desktop	180000	164000	150000	1	1.0
	9	desktop	1927	1825	900	1	1.0
	10	desktop	642	612	430	1	1.0

VI. Which products have the most purchases per campaign?

```
SELECT b.product_id AS "Product ID",  b.product_name AS "Product
Name", SUM(a.purchases) AS "Number of Products Sold Per Campign",
d.account_name AS "Account"
FROM campaign_results AS a , products AS b, campaigns As c, accounts AS
d
WHERE a.campaign_id= c.campaign_id
AND b.product_id = c.product_id
AND b.account_id = d.account_id
GROUP BY b.product_id
ORDER BY SUM(a.purchases) DESC;
```

	Product ID	Product Name	Number of Products Sold Per Camp...	Account
▶	10	Margaritas	1000	Serena
	50	Facials	600	NeuroSpa
	20	Dolls	185	ToySmart
	15	Shirt_fall_collection	100	SilviaTcherassi
	34	Games	50	BoingToys
	71	Rooms_ocean_view	9	HiltlonCaribbean
	33	Tacos	7	CoyoTaco
	27	Paintings	3	DesignDistrict
	30	Rooms	2	MoxyHotel
	66	Flights	1	TravelAir

VII. In which channels are our campaign performing the best?

```
SELECT ca.channel_type AS "Channel", SUM(c.purchases) as "Number of Purchases per Channel", c.Total_impressions, c.CTR, c.Total_reach
FROM Campaigns AS ca, Campaign_results AS c
WHERE ca.campaign_id = c.campaign_id
GROUP BY ca.channel_type, c.Total_impressions, c.CTR, c.Total_reach
ORDER BY COUNT(c.purchases) DESC;
```

Channel	Number of Purchases per Chan...	Total_impressions	CTR	Total_reach
facebook	1	2000	0.40	1390
YouTube	1	10000	0.06	8000
facebook	1	800	0.12	500
google	1	50000	0.34	20000
google	1	600	0.08	300
linkedin	1	700	8.57	75000
instagram	1	1500	0.03	1000
instagram	1	180000	0.83	164000
facebook	1	1927	0.46	1825
google	1	642	0.66	612

VIII. Does a higher investing in campaigns means a higher ROI?

```
SELECT a.creative_type, a.budget as 'Campaign Costs', a.campaign_name,
b.total_impressions, b.ROI, b.Clicks, b.purchases
from campaigns a, campaign_results b
where a.budget = (select max(a1.budget)     -- Don't forget the ()
                                                from campaigns a1
                  where a.campaign_id=a1.campaign_id)
and a.campaign_id = b.campaign_id
ORDER BY total_impressions DESC;
```

-

creative_type	Campaign Costs	campaign_name	total_impressio...	ROI	Clicks	purchases
display	500	margaritas_promotions_q1	180000	9.95	150000	1000
video	25000	facials_Q1	50000	8.15	17000	600
display	300	toys_dolls_Q1	10000	1.20	650	185
video	12208	BoingGames	2000	0.01	800	50
mobile	48829	tacos_promotions_q1	1927	4.03	900	7
video	6000	flights_q1	1500	0.05	37	1
display	15260	SVT_fall_collection	800	6.00	100	100
mobile	20000	rooms_ocean_views	700	5.62	60000	9
video	1250	discount_paintings_q1	642	1.50	430	3
native	12500	MOXY_rooms	600	3.01	50	2

- # PART 6 – DETAIL SQL PROCEDURES–

I.

The first built procedure will focus on that automatically updating the status of each campaign when they are due.
This procedure is meant to automatimize the process of storing the information regarding the status of a specific campaign without the help of a manual entry.
For the fact that some campaigns are about or still didn't start, the procedure is based on current date.

Input: end date of campaign as DATE

Functionality:

Step 1: declare a variable, v_status and v_end_date, to store the following query. Declared respectively as TINYTEXT and DATE

Step 2: selecting the previously declared variables and store them as the corresponding variables existing in the dataset

Step 3: If statement as:
IF v_end_date is equal to the current date then the status of the campaign is UPDATED and SET as SHUTDOWN.

Output: "SHUTDOWN" TINYTEXT
Please find below the code of the procedure we just described

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updatestatus`(IN in_end_date DATE ,OUT
out_status_upd tinytext)
BEGIN
DECLARE v_status TINYTEXT;
DECLARE v_end_date DATE;

select status, end_date
into v_status, v_end_date
from campaigns
where v_end_date = in_end_date;

if v_end_date = curdate() THEN
UPDATE campaigns
set v_status = "SHUTDOWN"
where v_status = ut_status_upd;
end if;
END
```

II.

As we mentioned in the first section, an effective omnichannel strategy is one of the most important attributes to measure and optimize campaigns. Therefore, there is an opportunity for Outsmart Labs to update the respective budget for a channel type that is proving more results on some campaigns that In other one.

This procedure will Identify all the campaigns id that have that channel type which is given as an input – look cur1 query. It select all the campaigns_id from campaigns that have the channel_type that was given as an input. Then the procedure updates the campaign objective to "conversions" for all those 'Facebook' campaign types (but it can be change to the channel the company wants to optimize). Lastly, in order to maximize results, the procedures updates the campaign budget by increasing it by 25% for the identified channel types.

Input: channel_type as varchar(45)

Functionality

Step 1 – Declare the variables needed to do the procedure and to store the following query. Declared v_campaign_id as varchar(10) and DECLARE C as varchar(10);

Step 2 – Select campaign_id of all campaigns of that channel type that was input

Step 3 – Declare cursor in order to handle the results set

Step 4 – Declare a NOT FOUND handler to handle the situation when the cursor could not find any row.

Step 5 – Open the cursor and open the loop

Step 6 – Inside the loop we used the finished variable

Step 7 – Begin the loop and fetch the statement with campaign_id

Step 8- Do exception handling. If there are no more records left, then exit the loop

Step 9 - Do the queries to update the objectives and the budget for the campaigns that have a channel type of "facebook". If the channel type is facebook, to update the objective for conversions and increase for 25% the budget.

Step 10 – End the if statements and close loop

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `check_Budget`(in in_channel_type
varchar(45))
BEGIN
/*Declare variables*/
DECLARE done INT DEFAULT FALSE;
DECLARE v_campaign_id varchar(10);
DECLARE C varchar(10);

 /*Select campaign_id of all campaigns the channel type */

 DECLARE cur1 CURSOR FOR
 select campaign_id
 from campaigns
 Where channel_type = in_channel_type;

 /* Exception handling for the cursor*/

 DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
 OPEN cur1;

 /*beginning of the loop*/

 read_loop: LOOP
 FETCH cur1 into v_campaign_id;

 /*Fetch the next record into variable v_campaign_id*/
 /*exception handling. If there are no more records left, then exit the loop. */

 IF done THEN
   LEAVE read_loop;
   END IF;

   /* Now I can do processing based on the value in v_campaign_id.*/
IF c= 'facebook' then
   update campaigns
   set objective = "conversions"
   where campaign_id = v_campaign_id
   and in_channel_type=c;
END IF;

IF c= 'facebook' then
   update campaigns
   set budget= 1.25*budget
   WHERE in_channel_type=c
```

```
AND campaign_id = v_campaign_id;
end if;
/*end of the loop*/
END LOOP;
END
```

Appendix

1. ER Model Script

```
-- -----------------------------------------------------
-- Schema team_project
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `team_project` DEFAULT CHARACTER SET utf8 ;
USE `team_project` ;

-- -----------------------------------------------------
-- Table `team_project`.`Campaigns`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Campaigns` (
  `campaign_id` INT NOT NULL,
  `campaign_name` VARCHAR(45) NULL DEFAULT NULL,
  `status` TINYTEXT NULL DEFAULT NULL,
  `start_date` DATE NULL DEFAULT NULL,
  `end_date` DATE NULL DEFAULT NULL,
  `Description` LONGTEXT NULL DEFAULT NULL,
  `channel_type` VARCHAR(45) NOT NULL,
  `budget` INT NULL DEFAULT NULL,
  `product_id` INT NULL DEFAULT NULL,
  `employee_id` INT NOT NULL,
  `Objective` VARCHAR(45) NULL DEFAULT NULL,
  `creative_type` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`campaign_id`, `channel_type`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Products`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Products` (
  `Product_id` INT NOT NULL,
  `campaign_id` VARCHAR(45) NULL DEFAULT NULL,
  `product_name` VARCHAR(45) NULL DEFAULT NULL,
  `product_link` VARCHAR(45) NULL DEFAULT NULL,
  `Account_id` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`Product_id`),
  CONSTRAINT `fk_Products_Campaigns1`
    FOREIGN KEY (`Product_id`)
```

```
    REFERENCES `team_project`.`Campaigns` (`product_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Invoices`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Invoices` (
  `invoice_id` INT NOT NULL,
  `invoice_amount` INT NOT NULL,
  `other_details` LONGTEXT NOT NULL,
  `account_id` INT NOT NULL,
  `invoice_issued` DATE NULL DEFAULT NULL,
  `STATUS` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`invoice_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Accounts`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Accounts` (
  `account_id` INT NOT NULL,
  `account_name` VARCHAR(45) NULL DEFAULT NULL,
  `account_address` VARCHAR(45) NULL DEFAULT NULL,
  `account_email` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`account_id`),
  CONSTRAINT `fk_Accounts_Products`
    FOREIGN KEY (`account_id`)
    REFERENCES `team_project`.`Products` (`Account_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Accounts_Invoices1`
    FOREIGN KEY (`account_id`)
    REFERENCES `team_project`.`Invoices` (`account_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

```
-- -----------------------------------------------------
-- Table `team_project`.`Agency_Employees`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Agency_Employees` (
  `employee_id` INT NOT NULL,
  `employee_name` VARCHAR(45) NULL DEFAULT NULL,
  `employee_last_name` VARCHAR(45) NULL DEFAULT NULL,
  `JobTitle` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`employee_id`),
  CONSTRAINT `fk_Agency_Employees_Campaigns1`
    FOREIGN KEY (`employee_id`)
    REFERENCES `team_project`.`Campaigns` (`employee_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Campaign_results`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Campaign_results` (
  `campaign_id` INT NOT NULL,
  `Total_reach` INT NOT NULL,
  `Total_engagement` INT NOT NULL,
  `Total_impressions` INT NOT NULL,
  `Purchases` INT NOT NULL,
  `ROI` DECIMAL(3,2) NOT NULL,
  `CTR` DECIMAL(3,2) NULL DEFAULT NULL,
  `Clicks` INT NULL DEFAULT NULL,
  PRIMARY KEY (`campaign_id`),
  CONSTRAINT `fk_Campaign_results_Campaigns1`
    FOREIGN KEY (`campaign_id`)
    REFERENCES `team_project`.`Campaigns` (`campaign_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Leads`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `team_project`.`Leads` (
  `user_id` INT NOT NULL,
  `user_name` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `Age` VARCHAR(45) NOT NULL,
  `Gender` VARCHAR(45) NOT NULL,
  `Country` VARCHAR(45) NOT NULL,
  `device` VARCHAR(45) NOT NULL,
  `campaign_id` INT NOT NULL,
  PRIMARY KEY (`user_id`),
  INDEX `fk_Leads_Campaigns1_idx` (`campaign_id` ASC) VISIBLE,
  CONSTRAINT `fk_Leads_Campaigns1`
    FOREIGN KEY (`campaign_id`)
    REFERENCES `team_project`.`Campaigns` (`campaign_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `team_project`.`Payments`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `team_project`.`Payments` (
  `Payment_id` INT NOT NULL,
  `Payment_Amount` INT NULL DEFAULT NULL,
  `other_details` LONGTEXT NULL DEFAULT NULL,
  `invoice_id` INT NOT NULL,
  PRIMARY KEY (`Payment_id`),
  INDEX `fk_Payments_Invoices1_idx` (`invoice_id` ASC) VISIBLE,
  CONSTRAINT `fk_Payments_Invoices1`
    FOREIGN KEY (`invoice_id`)
    REFERENCES `team_project`.`Invoices` (`invoice_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


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

2. Tables Data
   Note: The data was imported from an csv file into the SQL Import Wizard

Campaigns Table

```
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (1,'BoingGames','ACTIVE','2021-03-01','2021-04-04','boing_game','facebook',12208,34,1998,'conversions','video');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (2,'toys_dolls_Q1','SHUTDOWN','2021-03-01','2021-04-04','toysmart_campagin','YouTube',300,20,2535,'traffic','display');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (3,'SVT_fall_collection','ACTIVE','2021-01-01','2021-05-08','svt_campaing','facebook',15260,15,1990,'conversions','display');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (4,'facials_Q1','ACTIVE','2020-07-09','2021-08-09','facials_campaign','google',25000,50,6321,'conversions','video');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (5,'MOXY_rooms','SHUTDOWN','2020-09-10','2021-08-10','moxy_campaign','google',12500,30,4389,'conversions','native');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (6,'rooms_ocean_views','ACTIVE','2021-02-02','2022-09-09','rooms_campaign','linkedin',20000,71,1111,'conversions','mobile');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (7,'flights_q1','ACTIVE','2021-01-02','2021-05-04','gflights_campaign','instagram',6000,66,1693,'engagement','video');
INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`, `budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES
```

(8,'margaritas_promotions_q1','ACTIVE','2021-09-09','2021-06-01','margaritas_campaign','instagram',500,10,1990,'lead_generation','display');

INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`,`budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (9,'tacos_promotions_q1','ACTIVE','2021-09-09','2021-04-12','tacos_campaign','facebook',48829,33,3476,'conversions','mobile');

INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`,`budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (10,'discount_paintings_q1','SHUTDOWN','2021-03-01','2021-04-14','paintings_campaigns','google',1250,27,3476,'conversions','video');

INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`,`budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (11,'vacation_promotion','ACTIVE','2021-03-10','2021-04-10','moxy_promotions','google',2500,30,1998,'conversions','display');

INSERT INTO `` (`campaign_id`,`campaign_name`,`status`,`start_date`,`end_date`,`Description`,`channel_type`,`budget`,`product_id`,`employee_id`,`Objective`,`creative_type`) VALUES (12,'restaurant_promotion_Q1','ACTIVE','2021-03-12','2020-04-12','Coyo_Taco_digital_influencer_campagin','TikTok',8000,33,3476,'store_visits','display');

ACCOUNTS TABLE

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1009,'BoingToys','1st Education Street','BoingToys@gmail.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1220,'ToySmart','4n Canal Park','ToySmart@gmail.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1340,'SilviaTcherassi','1st Street','SVT@hotmail.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1360,'CoyoTaco','111 SW 1st Avenue','Coyotaco@restaurants.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1450,'DesignDistrict','1657 N Boston Avenue','District@designeres.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1580,'MoxyHotel','17315 Collins Ave ','Moxyhotel@gmail.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (1890,'TravelAir','19th Canal Park','Travelair@gmail.com');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (2020,'HiltlonCaribbean','31st Bonus Street','Hilton@student.hult.edu');

INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (5060,'NeuroSpa','1st Dalton Street','NeuroSpa@aol.com');

```
INSERT INTO `` (`account_id`,`account_name`,`account_address`,`account_email`) VALUES (6040,'Serena','240 Tremont St','Serena@moxyhotels.com');
```

Agency Employees

```
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (1111,'Marc','Friedman','Director of paid search',1);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (1489,'Gia','Lu','Marketing Coordinator',4);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (1693,'Mariana','Gutierrez','Art Director',3);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (1990,'Mariana','Ballesteros','Social media specialist',6);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (1998,'Tommaso','Mazzucco','CEO',1);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (2535,'Natalia','Gomez','Analyst',3);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (3476,'Alexandra','Rodriguez','Social media manager',9);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (3579,'David','Azar','SEO coordinator',7);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (3924,'Christian','Romero','Campaign Manager',5);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (4389,'Lidia','Calsamiglia','Account Manager',5);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (4664,'Cristina','Angel','Influencer Marketing Manager',2);
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (6190,'Rachael','Fredman','Marketing Specialist',2);
```

```
INSERT INTO `` (`employee_id`,`employee_name`,`employee_last_name`,`JobTitle`,`campaign_id`) VALUES (6321,'martha','Aycardi','Director of paid search',10);
```

Invoices

```
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (1078,25000,'advertising_march',1009,'2021-12-01','OVERDUE');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (1368,5000,'advertising_march',1450,'2021-03-01','UNPAID');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (1645,15000,'advertising_march',1360,'2021-03-03','PAID');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (2078,60000,'advertising_march',1580,'2021-03-01','PAID');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (2348,12900,'advertising_march',6040,'2021-03-01','OVERDUE');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (2575,10000,'advertising_march',1890,'2021-03-01','PAID');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (2681,8000,'advertising_march',2020,'2021-03-01','OVERDUE');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (4366,30000,'advertising_march',5060,'2021-03-01','UNPAID');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (4368,5000,'advertising_march',1220,'2021-03-01','OVERDUE');
INSERT INTO `` (`invoice_id`,`invoice_amount`,`other_details`,`account_id`,`invoice_issued`,`STATUS`) VALUES (9076,1000,'advertising_march',1340,'2021-03-01','PAID');
```

LEADS

```
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (1330,'Natalia Hernandez','natalia@gmail.com','31','F','Brazil','mobile',6);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (5672,'Julien Casablancas','julien@music.com','18','M','USA','desktop',4);
```

INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (5790,'Carly berry','carly@northeastern.edu','22','F','China','mobile',5);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (6566,'Natalia Castano','casta@hotmail.com','20','F','Peru','mobile',7);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (7658,' Natalia Gomez','natalia@hult.edu','25','F','Colombia','mobile',1);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (8091,'Tommas Mazzucco ','tommy@gmail.com','24','M','Italy','desktop',2);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (8309,'Marc Gomez','marc.132@aol.com','39','M','Germany','desktop',8);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (8589,'Cecicilia Brazil','cecilia@student.hult.edu','17','F','USA','desktop',10);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (9194,'Tuty Rose','tuty@hotmail.com','25','F','Spain','desktop',9);
INSERT INTO `` (`user_id`,`user_name`,`email`,`Age`,`Gender`,`Country`,`device`,`campaign_id`) VALUES (16660,'Ashley chica','ashley@harvard.edu','33','F','USA','mobile',3);

PAYMENTS

INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (1925,8000,'advertising_march',1078);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (1987,30000,'advertising_march',1368);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (2074,15000,'advertising_march',1645);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (2813,25000,'advertising_march',2078);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (4305,5000,'advertising_march',2348);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (4456,5000,'advertising_march',2575);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (5238,10000,'advertising_march',2681);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (7642,1000,'advertising_march',4366);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (7820,12900,'advertising_march',4368);
INSERT INTO `` (`Payment_id`,`Payment_Amount`,`other_details`,`invoice_id`) VALUES (8650,50000,'advertising_march',9076);

PRODUCTS

```
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (10,'8','Margaritas','https.margaritas.com','6040');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (15,'3','Shirt_fall_collection','https.shirts-and.coll.com','1340');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (20,'2','Dolls','https.dolls.com','1220');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (27,'10','Paintings','https.paint.com','1450');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (30,'5','Rooms','https.roomscom','1580');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (33,'9','Tacos','https.tacos.amo.','1360');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (34,'1','Games','https.games.com','1009');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (50,'4','Facials','https.facials.com','5060');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (66,'7','Flights','https.flights.com','1890');
INSERT INTO `` (`Product_id`,`campaign_id`,`product_name`,`product_link`,`Account_id`)
VALUES (71,'6','Rooms_ocean_view','https.rooms_ocean_view.com','2020');
```

CAMPAIGNS RESULTS

```
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (1,1390,1000,2000,50,0.01,0.40,800);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (2,8000,2400,10000,185,1.20,0.06,650);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (3,500,160,800,100,6.00,0.12,100);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (4,20000,5000,50000,600,8.15,0.34,17000);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (5,300,130,600,2,3.01,0.08,50);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (6,75000,46000,700,9,5.62,8.57,60000);
```

```
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (7,1000,67,1500,1,0.05,0.03,37);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (8,164000,145000,180000,1000,9.95,0.83,150000);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (9,1825,1200,1927,7,4.03,0.46,900);
INSERT INTO ``
(`campaign_id`,`Total_reach`,`Total_engagement`,`Total_impressions`,`Purchases`,`ROI`,`CTR`,`
Clicks`) VALUES (10,612,311,642,3,1.50,0.66,430);
```