

DATABASE PRINCIPLE PROJECT

Advertising Business Company Submission Director **(part2)**



TSUNG-TSE TU
S4780187

INDEX

1. Background

2. Query Demonstration

3. Conclusion

Background

Constructing an advertising demand system for clients to submit their requirement is an important approach for both clients and advertising service provider.

By providing a streamlined communication, the back-and-forth communication via other different channels (emails, phone calls, etc) could be eliminated to a reasonable degree, granting advertising agencies to quickly access the requirements and proceed with campaign planning and execution, reducing administrative overheads and allow both parties to focus on more value-added activities. It also gives clients to hand-in more detailed and comprehensive information of their advertising need, bringing accuracy and completeness for both ends.

For the module of the system, I implement the database based on the relational schema I created in part 1 and run a few queries and test if it works as intended.

Requirements:

Programming language: SQL

Software: 1. XAMPP: Enable to host and serve web, then create DB on it.

2.phpMyAdmin: handle administration of MYSQL over the web

Query Demonstration

- Join query

I have implemented join query between 'Client' and 'Request' table to show results of all the requests corresponding to their clients.

Both the table had the attribute Client_ID.

This join query is useful when trying to keep an eye on the client's request.

Table 1-Client

	Client_ID	Firstname	Lastname	Gender	Account name	Phone number	Birth date	Address
<input type="checkbox"/> Edit Copy Delete	C001	Charlie	Dedmon	Male	qwdqwd1678	48217291	1994-10-21	34 Cole Street
<input type="checkbox"/> Edit Copy Delete	C002	Keran	Stepheny	Female	qfqfqwd4567	481629272	2009-08-09	11 Heaven Street
<input type="checkbox"/> Edit Copy Delete	C003	Stephenie	Clarkson	Female	steph012	418264936	2005-07-17	666 Holy Street
<input type="checkbox"/> Edit Copy Delete	C004	Michael	Waskey	Male	mmm112	419272619	1996-01-30	1221 Ground Street
<input type="checkbox"/> Edit Copy Delete	C005	Yuri	Lin	Female	to1n1p23	418294712	1994-10-21	1 Courage Street

Table 2-Request

	Request_No	start date	status	Client_ID
<input type="checkbox"/> Edit Copy Delete	R001	2023-03-17	Pending	C003
<input type="checkbox"/> Edit Copy Delete	R002	2023-05-01	Confirmed	C003
<input type="checkbox"/> Edit Copy Delete	R003	2023-03-06	Rejected	C004
<input type="checkbox"/> Edit Copy Delete	R004	2023-05-06	Confirmed	C005

SQL CODE:

✓ Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
1 SELECT *
2 FROM client
3 LEFT JOIN request on client.Client_ID = request.Client_ID;
```

RESULT:

Client_ID	Firstname	Lastname	Gender	Account name	Phone number	Birth date	Address	Request_No	start date	status	Client_ID
C003	Stephenie	Clarkson	Female	steph012	418264936	2005-07-17	666 Holy Street	R001	2023-03-17	Pending	C003
C003	Stephenie	Clarkson	Female	steph012	418264936	2005-07-17	666 Holy Street	R002	2023-05-01	Confirmed	C003
C004	Michael	Waskey	Male	mmm112	419272619	1996-01-30	1221 Ground Street	R003	2023-03-06	Rejected	C004
C005	Yuri	Lin	Female	to1n1p23	418294712	1994-10-21	1 Courage Street	R004	2023-05-06	Confirmed	C005
C001	Charlie	Dedmon	Male	qwdqwd1678	48217291	1994-10-21	34 Cole Street	NULL	NULL	NULL	NULL
C002	Keran	Stepheny	Female	qfqfqwd4567	481629272	2009-08-09	11 Heaven Street	NULL	NULL	NULL	NULL

Shows NULL because not every client has to submit their request after their client index has been created.

- Division query

I used 'payroll record' and 'staff' table to showcase division query by retrieving staff members that hasn't been paid yet.

This query is usually good in retrieve records that satisfy a particular condition involving multiple tables.

Table 3-payroll record








				Payroll_No	SSN	date	money
<input type="checkbox"/>	 Edit	 Copy	 Delete	P001	S003	2023-04-01	10000
<input type="checkbox"/>	 Edit	 Copy	 Delete	P002	S002	2023-04-01	23500
<input type="checkbox"/>	 Edit	 Copy	 Delete	P003	S004	2023-04-01	7500
<input type="checkbox"/>	 Edit	 Copy	 Delete	P004	S005	2023-04-01	13000
<input type="checkbox"/>	 Edit	 Copy	 Delete	P005	S002	2023-05-01	25000

Table 4- Staff

SSN	Gender	Account name	Birth date	Address	Phone number	Firstname	Lastname	Dept_No	start date
S001	Male	zzz123	1993-05-19	62 Butler Crescent	418291729	George	Paul	D001	2023-01-04
S002	Male	dqwdqw12	1994-12-19	60 Cunningham Street	491570157	Kevin	Durant	D001	2013-07-13
S003	Female	steph012	1999-01-12	82 Flinstone Drive	418264936	Emma	Watson	D004	2016-02-08
S004	Female	mmm112	1987-02-28	60 Settlement Road	419272619	Daisy	Williams	D002	1999-05-22
S005	Male	feefbe689	1973-11-17	58 Goebels Road	471973569	Tony	Allen	D003	1997-06-27

SQL CODE:




Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)

```

SELECT SSN, Firstname, Lastname FROM staff WHERE SSN NOT IN ( SELECT SSN FROM `payroll record` );
    
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

RESULT:

<div>↔ T ↔</div>				SSN	Firstname	Lastname
<input type="checkbox"/>	 Edit	 Copy	 Delete	S001	George	Paul

Shows that one employee hasn't been paid yet.

- Update operation

I implemented the update query to change status on the client's request or the current project status, here we take 'project' table for example.

This query is good for maintaining control of the works we're handling.

Table 5- Project (before update)

Project_No	start date	due date	status	budget	Request_No	Dept_No
P001	2023-02-15	2023-05-15	Working on	25000	R004	D002
P002	2023-05-08	2023-05-21	Finished	100000	R002	D003

SQL CODE:

✓ 1 row affected. (Query took 0.0010 seconds.)

```
UPDATE project SET status = 'finish' WHERE Project_No = 'P001' AND status = 'working on';
```

RESULT:

Project_No	start date	due date	status	budget	Request_No	Dept_No
P001	2023-02-15	2023-05-15	finish	25000	R004	D002
P002	2023-05-08	2023-05-21	Finished	100000	R002	D003

Status for P001 project changed to finish.

- Aggregation query (SUM)

I implemented aggregation query to find out the total payroll for each staff.

Table 6 - Payroll record

Payroll_No	SSN	date	money
P001	S003	2023-04-01	10000
P002	S002	2023-04-01	23500
P003	S004	2023-04-01	7500
P004	S005	2023-04-01	13000
P005	S002	2023-05-01	25000

SQL CODE:

```
✓ Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

SELECT SSN, SUM(money) AS total_payroll FROM `payroll record` GROUP BY SSN;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

RESULT:

SSN	total_payroll
S002	48500
S003	10000
S004	7500
S005	13000

Prints out every staffs' total payroll (S001 has not been paid).

- Aggregation with group-by (aggregated value for each group)

For this aggregation query, I will try to count the days from start date to due date and group by request, it could show how much time the ad agency had to finish client's demand.

Table 7- Project(aggregate)

Project_No	start date	due date	status	budget	Request_No	Dept_No
P001	2023-02-15	2023-05-15	finish	25000	R004	D002
P002	2023-05-08	2023-05-21	Finished	100000	R002	D003

SQL CODE:

✓ Showing rows 0 - 1 (2 total, Query took 0.0009 seconds.)

```
SELECT Request_No, DATEDIFF(`due date`, `start date`) AS days_count FROM project GROUP BY Request_No;
```

RESULT:

Request_No	days_count
R002	13
R004	89

Shows the days left to finish client request.

Conclusion

For this project, I spent nearly 15 hours non-stop on this whole journey. Since I spent a lot of time working on the first part (most of the time were spent on changing the ER diagram repeatedly), the second part wasn't too difficult to start with, which shows the importance of designing the relational schema of the database.

I really like the part when you find out your design works after you setup the environment (XAMPP and phpMyAdmin) and implement all the stuff in, then you get to mess around with it. It still hasn't reached the potential consider that it's only the initial stage and still had a lot of improvements to be done such as designing a user interface, or try using hash functions to store confidential information, but this project really helps me understand the usage of SQL and what it takes to create a database beforehand.

If you want to have a steady pace from the beginning, I recommend that you **DON'T** do your work straight way, but try to have a clear plan and scope of what you wanted to demonstrate first, in other word, focus on the brainstorming stage; come up with practical ideas and put yourself in the user position, think of all the pros and cons.