

Exploring SQL

Week 3 - ITC 6000

College of Professional Studies

Northeastern University - Vancouver

Representatives

Smit Parmar

Abhilash Dikshit

Shamim Sherafati

Nastaran Zamanian

Abstract

In the previous modules, we developed understanding of ERD and SQL Language. After we understand how the entities and attributes can be linked through an ERD and how to create relational databases, we shall start with a deeper understanding of the SQL Language. Hence in this weekly project we shall be using some functions to derive certain outputs from our database. This will help us grasp the practical understanding of those functions and improve our skills in SQL. Thereby, this report is duly submitted by Group 2 of ITC 6000 : Database Management Systems and is submitted before the provided date and time.

Introduction

In Week 2, we created a Database named 'bank_ca' which had 6 Entities and every entity had multiple attributes. For the table bank, we had the attributes 'ba_name', 'ba_code', 'ba_address' and 'ba_type'. For the table branch, we had the attributes 'b_id', 'b_address' and 'b_name'. For the table Loan we had the attributes 'loan_type', 'loan_id', 'loan_amount', 'loan_duration' and 'loan_interest'. For the table Account, we had the attributes 'acc_no', 'acc_balance' and 'acc_type'. For the table Customer, we had the attributes 'c_id', 'c_name', 'c_dob', 'c_contact' and 'c_address'. For the table Employee, we had the attributes 'e_id', 'e_name', 'e_dob', 'e_dept', 'e_contact', 'e_address' and 'e_age'. After creating the tables with the respective attributes, we inserted values inside our tables.

Method

The table below shows the original bank table.

	ba_code	ba_name	ba_address	ba_type
1	1	Royal National Bank	1350 Georgia Street	Public
2	2	CIBC	555 Robson Street	Public
3	3	Canadian Western Bank	436 Duismuir Street	Private
4	4	National Bank of Canada	324 Homer Street	Private
5	5	Exchange Bank of Canada	245 Georgia Street	Public
6	6	BMO Financial Group	177 West Cordova Street	Private
7	7	B2B Bank	777 Richards Street	Public
8	8	The Bank of Nova Scotia	458 Granville Street	Private
9	9	Fairstone Bank of Canada	131 Burrard Street	Private
10	10	RFA Bank of Canada	439 Robson Street	Public

We want to add some new rows to the current table. So, we run the following code.

```
INSERT INTO bank VALUES
(11, 'Sciotia Bank', 'West Georgia Street','Public'),
(12, 'CIBC Bank', 'West Georgia Street','Private'),
(13, 'TD Bank', 'West Georgia Street','Public');
```

The output table is shown in the following. As it can be seen the number of its rows increases from 10 to 13.

Results		Messages			
	ba_code	ba_name	ba_address	ba_type	
1	1	Royal Nat...	1350 Georgia ...	Public	
2	2	CIBC	555 Robson St...	Public	
3	3	Canadian ...	436 Duismuir ...	Private	
4	4	National ...	324 Homer Str...	Private	
5	5	Exchange ...	245 Georgia S...	Public	
6	6	BMO Finan...	177 West Cord...	Private	
7	7	B2B Bank	777 Richards ...	Public	
8	8	The Bank ...	458 Granville...	Private	
9	9	Fairstone...	131 Burrard S...	Private	
1...	10	RFA Bank ...	439 Robson St...	Public	
1...	11	Sciotia B...	West Georgia ...	Public	
1...	12	CIBC Bank	West Georgia ...	Private	
1...	13	TD Bank	West Georgia ...	Public	

Since there is a wrong spelling with the name of Sciotia bank in ba_name column in 11th row, we need to correct it with the help of UPDATE command like below.

```
UPDATE bank SET ba_name='Scotia Bank' WHERE ba_code=11;
```

Here we can see the change in the bank table.

Results Messages				
	ba_code	ba_name	ba_address	ba_type
1	1	Royal National Bank	1350 Georgia Street	Public
2	2	CIBC	555 Robson Street	Public
3	3	Canadian Western Bank	436 Duismuir Street	Private
4	4	National Bank of Canada	324 Homer Street	Private
5	5	Exchange Bank of Canada	245 Georgia Street	Public
6	6	BM0 Financial Group	177 West Cordova Street	Private
7	7	B2B Bank	777 Richards Street	Public
8	8	The Bank of Nova Scotia	458 Granville Street	Private
9	9	Fairstone Bank of Canada	131 Burrard Street	Private
10	10	RFA Bank of Canada	439 Robson Street	Public
11	11	Scotia Bank	West Georgia Street	Public
12	12	CIBC Bank	West Georgia Street	Private
13	13	TD Bank	West Georgia Street	Public

The original table for loan is shown in the following.

	loan_type	loan_id	loan_amount	loan_duration	loan_interest	c_id	b_id
1	Debt Consolidation	426436795	140000	3	10.60	983	B2739
2	Personal	426436789	50000	12	3.40	2130	RN235
3	Credit-Builder	426436794	100000	6	5.30	2345	BM378
4	Home Equity	426436793	54000	36	9.35	4200	EB783
5	Student	426436797	20000	12	12.99	4702	F6793
6	Student	426436791	40000	24	12.99	5348	CW275
7	Mortgage	426436798	900000	68	3.01	6018	A4D67
8	Payday	426436796	15000	1	8.40	6721	NSC34
9	Auto	426436790	55000	24	5.28	7658	C6778
10	Mortgage	426436792	90000	56	3.01	7891	NB870

We want to make some modifications in this table for the amount of interest in which our condition holds. To this end we should use UPDATE command in which each restriction is shown.

UPDATE loan SET loan_interest=15 WHERE loan_type='Mortgage';

UPDATE loan SET loan_interest=10 WHERE loan_type='Credit-Builder';

UPDATE loan SET loan_interest=8 WHERE loan_type='Home Equity';

According to the output we can see these changes. For example, the interest of Mortgage change to 15.

Create Insight

loan_type	loan_id	loan_amo...	loan_durat...	loan_inter...	c_id	b_id
Debt Cons...	426436795	140000	3	10.60	983	B2739
Personal	426436789	50000	12	3.40	2130	RN235
Credit-Buil...	426436794	100000	6	10.00	2345	BM378
Home Equity	426436793	54000	36	8.00	4200	EB783
Student	426436797	20000	12	12.99	4702	F6793
Student	426436791	40000	24	12.99	5348	CW275
Mortgage	426436798	900000	68	15.00	6018	A4D67
Payday	426436796	15000	1	8.40	6721	NSC34
Auto	426436790	55000	24	5.28	7658	C6778
Mortgage	426436792	90000	56	15.00	7891	NB870

In the employee table we can see wrong age when comparing related date of birth. Take row 1, according to date of birth, age should be 20 rather than 23. Hence, we should correct the data. We do it for whole values in this column. Below shows the code and related table.

UPDATE employee SET e_age = DATEDIFF(MONTH,e_dob,GETDATE())/12;

Results		Messages	Query Plan	Plan Tree	Top Operations		
	e_id	e_name	e_dob	e_dept	e_contact	e_address	e_age
1	984	Jack Brown	2002-05-17	HR	9081451073	52nd Avenue Surrey	20
2	2131	Smit Parmar	2000-07-20	Investment Planning	21846570194	177 Robson Street	22
3	2346	John Smith	1999-03-01	Marketing	4658962931	West Vancouver	23
4	4201	Sarah Nadi	1989-11-06	Investment Planning	8971256891	781 Duismuir Street	33
5	4703	Dave Park	1970-04-10	Corporate Services	9608512278	256 Metrotown	52
6	5349	Shamim Sherafati	1997-10-28	HR	687213456	768 Richards Street	25
7	6019	Justin Bieber	1994-08-29	Marketing	6354387654	2nd Avenue Surrey	28
8	6722	Jacquile Fernandez	1989-01-12	Investment Planning	7420638906	555 Seymour Street	33
9	7659	Abhilash Dikshit	1993-05-10	Marketing	2365744375	W 23 Cordova Street	29
1...	7892	Nastaran Zamanian	1992-01-16	Corporate Services	1879578123	West Vancouver	30

This table shows values before the mentioned modifications.

	e_id	e_name	e_dob	e_dept	e_contact	e_address	e_age
1	984	Jack Brown	2002-05-17	HR	9081451073	52nd Avenue Surrey	23
2	2131	Smit Parmar	2000-07-20	Investment Planning	21846570194	177 Robson Street	21
3	2346	John Smith	1999-03-01	Marketing	4658962931	West Vancouver	39
4	4201	Sarah Nadi	1989-11-06	Investment Planning	8971256891	781 Duismuir Street	48
5	4703	Dave Park	1970-04-10	Corporate Services	9608512278	256 Metrotown	52
6	5349	Shamim Sherafati	1997-10-28	HR	687213456	768 Richards Street	25
7	6019	Justin Bieber	1994-08-29	Marketing	6354387654	2nd Avenue Surrey	22
8	6722	Jacquile Fernandez	1989-01-12	Investment Planning	7420638906	555 Seymour Street	33
9	7659	Abhilash Dikshit	1993-05-10	Marketing	2365744375	W 23 Cordova Street	29
10	7892	Nastaran Zamanian	1992-01-16	Corporate Services	1879578123	West Vancouver	30

Now we want to delete those rows whose loan_id is null.

DELETE FROM loan **WHERE**[loan_id] **IS NULL**

Results Messages Query Plan Plan Tree Top Operations							
	loan_type	loan_id	loan_amount	loan_duration	loan_interest	c_id	b_id
1	Debt Consolidation	426436795	140000	3	10.60	983	B2739
2	Personal	426436789	50000	12	3.40	2130	RN235
3	Credit-Builder	426436794	100000	6	10.00	2345	BM378
4	Home Equity	426436793	54000	36	8.00	4200	EB783
5	Student	426436797	20000	12	12.99	4702	F6793
6	Student	426436791	40000	24	12.99	5348	CW275
7	Mortgage	426436798	900000	68	15.00	6018	A4D67
8	Payday	426436796	15000	1	8.40	6721	NSC34
9	Auto	426436790	55000	24	5.28	7658	C6778
1...	Mortgage	426436792	90000	56	15.00	7891	NB870

The output does not show any changes when comparing to the previous loan table. Because the table does not include any empty value.

In the next step we want to delete the 3 rows added before in the bank table.

DELETE FROM bank **WHERE** ba_code **BETWEEN** 11 **AND** 13;

Results	Messages	Query Plan	Plan Tree	Top Operations
	ba_code	ba_name	ba_address	ba_type
1	1	Royal National Bank	1350 Georgia Street	Public
2	2	CIBC	555 Robson Street	Public
3	3	Canadian Western Bank	436 Duismuir Street	Private
4	4	National Bank of Canada	324 Homer Street	Private
5	5	Exchange Bank of Canada	245 Georgia Street	Public
6	6	BMO Financial Group	177 West Cordova Street	Private
7	7	B2B Bank	777 Richards Street	Public
8	8	The Bank of Nova Scotia	458 Granville Street	Private
9	9	Fairstone Bank of Canada	131 Burrard Street	Private
1...	10	RFA Bank of Canada	439 Robson Street	Public

We intend to define a new age column in the customer table. At first it has null values for all columns. Then with the help of UPDATE command we set a value for each customer by using date of birth.

ALTER TABLE customer **ADD** c_age **INT**

Results		Messages	Query Plan	Plan Tree	Top Operations		
	c_id	c_name	c_dob	c_contact	c_address	c_age	
1	983	Jack Brown	2002-05-17	9081451073	52nd Avenue Surrey	NULL	
2	2130	Smit Parmar	2000-07-20	21846570194	177 Robson Street	NULL	
3	2345	John Smith	1999-03-01	4658962931	West Vancouver	NULL	
4	4200	Sarah Nadi	1989-11-06	8971256891	781 Duismuir Street	NULL	
5	4702	Dave Park	1970-04-10	9608512278	256 Metrotown	NULL	
6	5348	Shamim Sherafati	1997-10-28	687213456	768 Richards Street	NULL	
7	6018	Justin Bieber	1994-08-29	6354387654	2nd Avenue Surrey	NULL	
8	6721	Jacquile Fernandez	1989-01-12	7420638906	555 Seymour Street	NULL	
9	7658	Abhilash Dikshit	1993-05-10	2365744375	W 23 Cordova Street	NULL	
1...	7891	Nastaran Zamanian	1992-01-16	1879578123	West Vancouver	NULL	

UPDATE customer SET c_age = DATEDIFF(MONTH,c_dob,GETDATE())/12;

Results		Messages	Query Plan	Plan Tree	Top Operations		
	c_id	c_name	c_dob	c_contact	c_address	c_age	
1	983	Jack Brown	2002-05-17	9081451073	52nd Avenue Surrey	20	
2	2130	Smit Parmar	2000-07-20	21846570194	177 Robson Street	22	
3	2345	John Smith	1999-03-01	4658962931	West Vancouver	23	
4	4200	Sarah Nadi	1989-11-06	8971256891	781 Duismuir Street	33	
5	4702	Dave Park	1970-04-10	9608512278	256 Metrotown	52	
6	5348	Shamim Sherafati	1997-10-28	687213456	768 Richards Street	25	
7	6018	Justin Bieber	1994-08-29	6354387654	2nd Avenue Surrey	28	
8	6721	Jacquile Fernandez	1989-01-12	7420638906	555 Seymour Street	33	
9	7658	Abhilash Dikshit	1993-05-10	2365744375	W 23 Cordova Street	29	
1...	7891	Nastaran Zamanian	1992-01-16	1879578123	West Vancouver	30	

Now we want drop the column created for age.

ALTER TABLE customer DROP COLUMN c_age

Results		Messages	Query Plan	Plan Tree	Top Operations					
	c_id	▼	c_name	▼	c_dob	▼	c_contact	▼	c_address	▼
1	983		Jack Brown		2002-05-17		9081451073		52nd Avenue Surrey	
2	2130		Smit Parmar		2000-07-20		21846570194		177 Robson Street	
3	2345		John Smith		1999-03-01		4658962931		West Vancouver	
4	4200		Sarah Nadi		1989-11-06		8971256891		781 Duismuir Street	
5	4702		Dave Park		1970-04-10		9608512278		256 Metrotown	
6	5348		Shamim Sherafati		1997-10-28		687213456		768 Richards Street	
7	6018		Justin Bieber		1994-08-29		6354387654		2nd Avenue Surrey	
8	6721		Jacquile Fernandez		1989-01-12		7420638906		555 Seymour Street	
9	7658		Abhilash Dikshit		1993-05-10		2365744375		W 23 Cordova Street	
1...	7891		Nastaran Zamanian		1992-01-16		1879578123		West Vancouver	

We use the following commands for changing the type of data. It gives the data more space for storing. Now c_name and e_name have VARCHAR(30) type, double the space compared to previous one NVARCHAR(15).

ALTER TABLE customer ALTER COLUMN c_name VARCHAR(30)

ALTER TABLE employee ALTER COLUMN e_name VARCHAR(30)

For understanding the max and min age of employee we run the code below. So the oldest and youngest workers are 52 and 20 respectively.

```
SELECT MAX(e_age) FROM employee;
```

Results		Messages	Query P
	(No column name)	▼	
1	52		

In addition, some max and min functions run for 2 tables of account and loan. Their results as a number are displayed here.

```
SELECT MIN(e_age) FROM employee;
```

Results		Messages	Query P
	(No column name)	▼	
1	20		

Accordingly max and min amount of account balance is 200000 and 2000 respectively.

```
SELECT MAX(acc_balance) FROM account;
```

Results		Messages	Query P
	(No column name)	▼	
1	2000000		

`SELECT MIN(acc_balance) FROM account;`

Results		Messages	Query P
	(No column name)	▼	
1	2000		

Here we get the max and min amount of loan which is given by banks.

`SELECT MAX(loan_amount) FROM loan;`

Results		Messages	Query P
	(No column name)	▼	
1	900000		

```
SELECT MIN(loan_amount) FROM loan;
```

Results		Messages	Query P
	(No column name)	▼	
1	15000		

Here we use 'Sum Function' for getting the sum of loan, employee, and customer.

```
SELECT SUM(loan_amount) AS 'SUM' FROM loan;
```

	SUM	▼
1	1464000	

```
SELECT SUM(loan_interest) AS 'SUM' FROM loan;
```

	SUM	▼
1	101.66	

As we can see, the sum of loan amount is 1464000 and the sum of loan interest is lower than that with 101.66.

Now COUNT FUNCTION WITH DISTINCT VALUES.

```
SELECT COUNT(DISTINCT ba_type) FROM bank;
```

	(No column name) ▾
1	2

```
SELECT COUNT(DISTINCT acc_type) FROM account;
```

	(No column name) ▾
1	8


```
SELECT COUNT(DISTINCT e_id) FROM employee;
```

	(No column name) ▾
1	10


The count function which we used to know the total number of rows in the orders table, give us the information about bank type which we have two types. In account type it shows 8 types and also for employee id, it counts 10.

Now we get the AVERAGE with average FUNCTION.


```
SELECT AVG(acc_balance) AS 'AVERAGE' FROM account;
```

	AVERAGE 
1	237900.000000

`SELECT AVG(loan_amount) AS 'AVERAGE' FROM loan;`

	AVERAGE 
1	146400.000000

`SELECT AVG(e_age) AS 'AVERAGE' FROM employee;`

	AVERAGE 
1	29

From the results above, it shows the average of account balance with 237900, for loan amount and employee age it shows 146400 and 29 respectively which depicts that the average age of employees working in a bank is 29 years old. 146,400 in average amount of loan given to people with an average of 237,900 balance on their loan.

SUMMARY

In the following Project we used the UPDATE command to alter the existing values in our table. We also dropped certain columns from our tables using the ALTER TABLE and DROP command. We also used ALTER TABLE to change the data types of certain attributes. We were also able to find the minimum and maximum values in a column using the SELECT MIN and SELECT MAX function. Next we calculated the SUM, COUNT and AVERAGE as well using the SELECT Function.

References

[1] (2021). *SQL Commands*. Codecademy.com.

<https://www.codecademy.com/article/sql-commands>

[2] (2018). *SQL Command Reference*. Docs.snowflake.com.

<https://docs.snowflake.com/en/sql-reference-commands.html>

[3] R, J. (2016). *The SQL Select Statement*. W3schools.com.

https://www.w3schools.com/sql/sql_select.asp

[4] (2020). *SQL INSERT, UPDATE, and DELETE Commands*. w3schools.com.

https://infocenter.informationbuilders.com/wf80/index.jsp?topic=%2Fpubdocs%2Freporting%2FCreatingReportswithWFLanguage%2Fsource%2FSQL_INSERT.htm

[5] Seattler, M. (2020). *SQL Aggregate Functions*. udacity.com.

<https://www.udacity.com/blog/2021/04/sql-aggregate-functions-avg-count-distinct-max-min-sum.html>