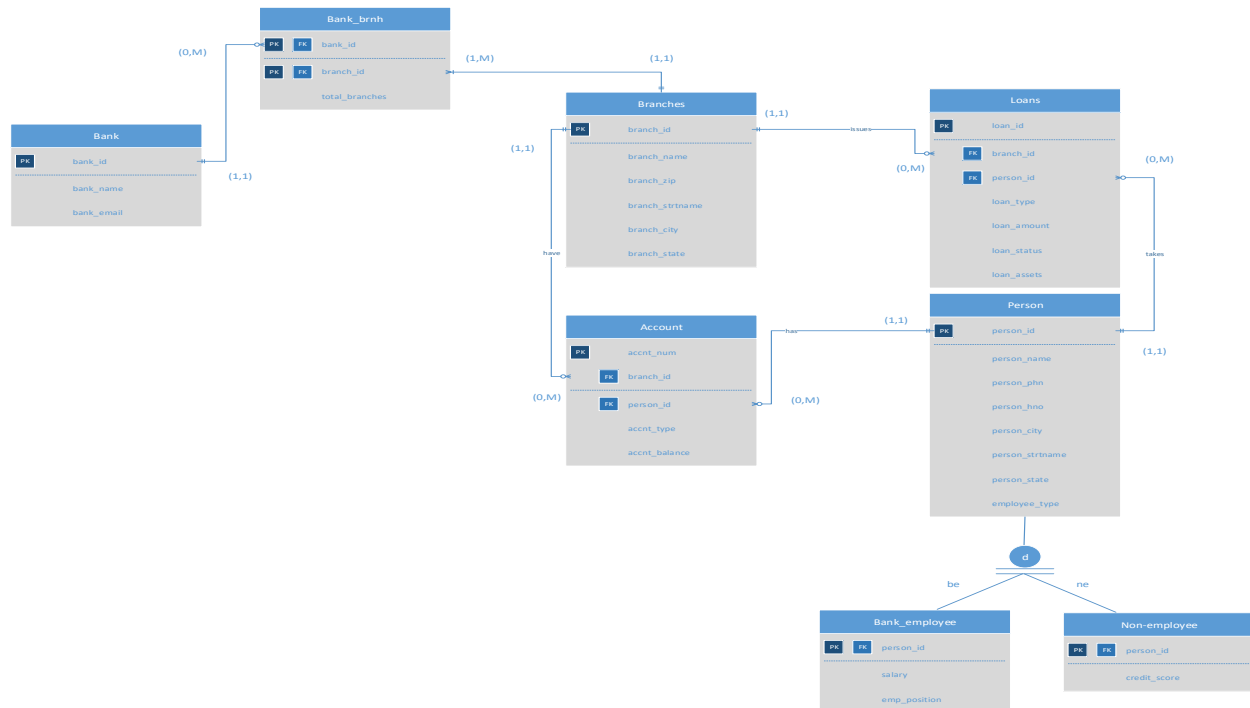


AIT-524-002  
Database Management Systems  
Assignment-11  
SUJAN CHAVA  
George Mason University

**ER- Diagram:**

Write four SQL queries to join two or more tables where each query contains multiple nested single-row functions. Make sure to use different functions in each query. Explain what each query is intended to do.

- 1) Display the names of the bank and city of the branches each bank has, where each word in a bank name and city of the branches of bank, starts with a capital letter and all other letters are lower case.

Worksheet Query Builder

```
select INITCAP(LOWER(bank_name)) bank_name, INITCAP(LOWER(branch_city)) city_of_branch from bank JOIN bank_brnh USING(bank_id)
JOIN branches USING(branch_id)
```

SUJAN CHAVA

---

Query Result x

All Rows Fetched: 3 in 0.328 seconds

BANK_NAME	CITY_OF_BRANCH
1 Bank Of America Fairfax	
2 Bank Of America Baltimore	
3 Pnc Bank	Fort Worth

- 2) Display the name and street name of the person, amount of loan taken by the person, street name of the branch and city of branch from which loan was taken, where word 'street' in street name of branch is replaced by 'strt' and entire street name of branch is displayed in uppercase.

Worksheet Query Builder

```
select person_name, person_strtname, loan_amount, UPPER(REPLACE(branch_strtname, 'street', 'strt')) branch_street, branch_city from person
JOIN loans USING(person_id) JOIN branches USING(branch_id)
```

SUJAN CHAVA

Query Result x

SQL | All Rows Fetched: 3 in 0.016 seconds

	PERSON_NAME	PERSON_STRTNAME	LOAN_AMOUNT	BRANCH_STREET	BRANCH_CITY
1	Venkat Annula	Ramona Dr	100000	MAIN STRT	fairfax
2	Sujan Chava	Freemont Strt	50000	RAMONA STRT	Baltimore
3	Vamsi Gunala	Shirle Dr	5000	ROYAL STRT	Fort worth

- 3) Display the name of the branch, address of the person who have bank accounts and length of the address of each person. NOTE: Here address includes house number, street name, city and state of the person.

Worksheet Query Builder

```
select branch_name, CONCAT(person_hno, CONCAT(person_strtname, CONCAT(person_city, person_state))) address,
LENGTH(CONCAT(person_hno, CONCAT(person_strtname, CONCAT(person_city, person_state)))) length_of_address
from person JOIN account USING(person_id) JOIN branches USING(branch_id)
```

SUJAN CHAVA

Query Result x

SQL | All Rows Fetched: 3 in 0.059 seconds

	BRANCH_NAME	ADDRESS	LENGTH_OF_ADDRESS
1	Fairfax@bofa	H:NO- 4307Ramona DrFairfaxVirginia	34
2	Balti@bofa	H:NO- 2071Shirle DrHerndonVirginia	34
3	PNC@Fort	H:NO- 4303Freemont StrtLas VegasNevada	38

- 4) Display the first name of bank employees and employee position. (Assume we doesn't know the length of name, the first and last names are separated by a space).

Worksheet Query Builder

```
select SUBSTR(person_name,1,INSTR(person_name,' ')) first_name, emp_position from person JOIN bank_employee USING(person_id)
```

SUJAN CHAVA

Query Result x

SQL | All Rows Fetched: 3 in 0.015 seconds

	FIRST_NAME	EMP_POSITION
1	Venkat	Receptionist
2	Vamsi	Manager
3	Sujan	Accountant

- 5) Write a query which is based on a single table and includes the GROUP BY statement. Explain what the query is supposed to do.

Retrieve the id of the person and total balance in all the accounts of each person.

Worksheet Query Builder

```
select person_id, SUM(ACCNT_BALANCE) total_balance from account group by (person_id)
```

SUJAN CHAVA

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.069 seconds

	PERSON_ID	TOTAL_BALANCE
1	221	2000
2	111	14000
3	211	9000

- 6) Repeat problem 5 but add the WHERE statement. Explain what the query is supposed to do.

Retrieve the total balance in all the accounts of person with the id 111 and 211.

The screenshot shows a SQL Query Builder window with a worksheet tab. The query text is: `select person_id, SUM(ACCT_BALANCE) total balance from account where person_id IN(111,211) group by(person_id)`. Below the query, the name "SUJAN CHAVA" is visible. The bottom panel shows the "Query Result" tab with a table of results.

PERSON_ID	TOTAL_BALANCE	
1	111	14000
2	211	9000

- 7) Repeat problem 6 but add the HAVING statement. Explain what the query is supposed to do.

Retrieve the total balance in all the accounts of person with the id 111 and 211 where total balance is greater than 10000.

The screenshot shows a SQL Query Builder window with a worksheet tab. The query text is: `select person_id, SUM(ACCT_BALANCE) total balance from account where person_id IN(111,211) group by(person_id) having SUM(ACCT_BALANCE)>10000`. Below the query, the name "SUJAN CHAVA" is visible. The bottom panel shows the "Query Result" tab with a table of results.

PERSON_ID	TOTAL_BALANCE	
1	111	14000

- 8) Write a query which is based on three tables and includes the GROUP BY, WHERE and HAVING statements. Explain what the query is supposed to do.

Retrieve the person name, and total balance in their accounts who have applied for loan amount greater than 5000 and having total account balance less than 5000.

Worksheet Query Builder

```
select person name,SUM(accnt balance) total_balance from person JOIN account USING(person_id) JOIN loans USING(person_id)
where loan_amount>5000 group by(person_name) having SUM(accnt_balance)<5000;
```

SUJAN CHAVA

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.791 seconds

PERSON_NAME	TOTAL_BALANCE
1 Sujan Chava	2000

- 9) Write an SQL query that uses the GROUP BY ROLL statement. Explain what the query is supposed to do.

Retrieve the id of the person, type of account and subtotals of account balance based on both id of the person and type of account, also list out the subtotals based on id of the person and grand total.

Worksheet Query Builder

```
select person_id,acct_type,SUM(accnt_balance) total_balance from account group by ROLLUP(person_id,acct_type);
```

SUJAN CHAVA

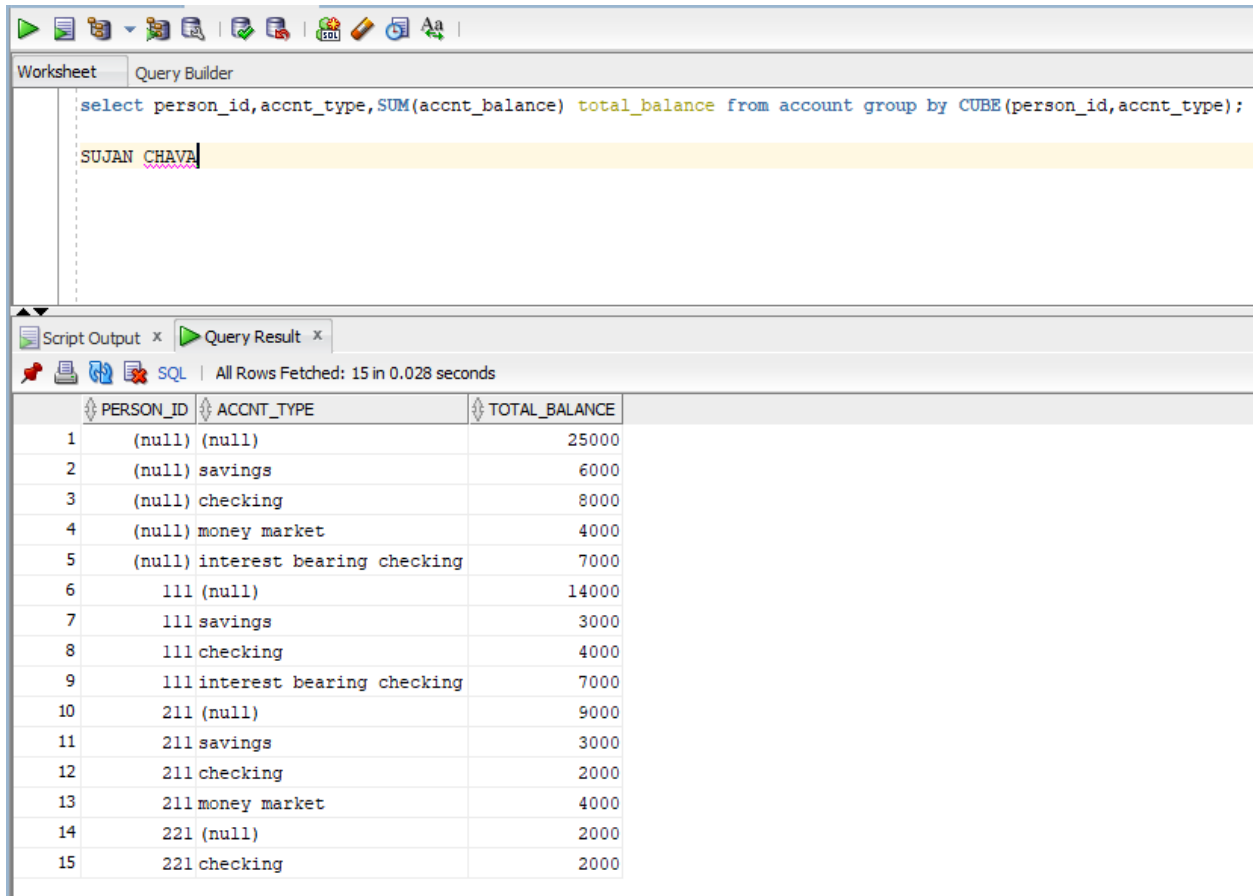
Script Output x Query Result x

SQL | All Rows Fetched: 11 in 0.02 seconds

PERSON_ID	ACCT_TYPE	TOTAL_BALANCE
1	111 savings	3000
2	111 checking	4000
3	111 interest bearing checking	7000
4	111 (null)	14000
5	211 savings	3000
6	211 checking	2000
7	211 money market	4000
8	211 (null)	9000
9	221 checking	2000
10	221 (null)	2000
11	(null) (null)	25000

- 10) Write an SQL query that uses the GROUP BY CUBE statement. Explain what the query is supposed to do.

Retrieve all the possible subtotals of account balance based on the id of the person and type of the account along with id of the person and type of the account.



The screenshot shows a SQL query editor with the following query:

```
select person_id,acct_type,SUM(acct_balance) total_balance from account group by CUBE(person_id,acct_type);
```

The query results are displayed in a table with 15 rows. The columns are PERSON\_ID, ACCNT\_TYPE, and TOTAL\_BALANCE.

	PERSON_ID	ACCNT_TYPE	TOTAL_BALANCE
1	(null)	(null)	25000
2	(null)	savings	6000
3	(null)	checking	8000
4	(null)	money market	4000
5	(null)	interest bearing checking	7000
6	111	(null)	14000
7	111	savings	3000
8	111	checking	4000
9	111	interest bearing checking	7000
10	211	(null)	9000
11	211	savings	3000
12	211	checking	2000
13	211	money market	4000
14	221	(null)	2000
15	221	checking	2000