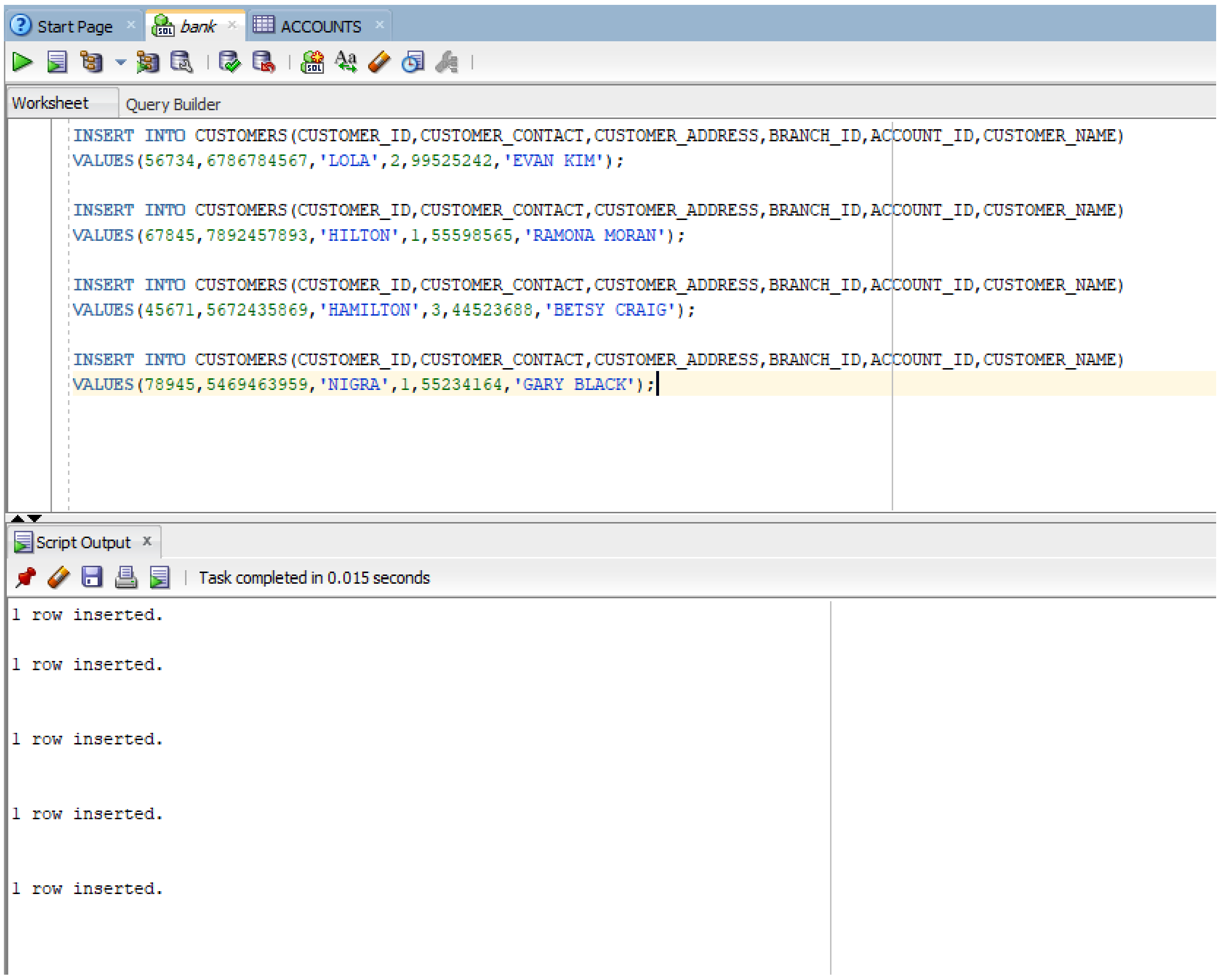
SQL Designing Views/Simple Queries

CREATE CUSTOMER TABLE:

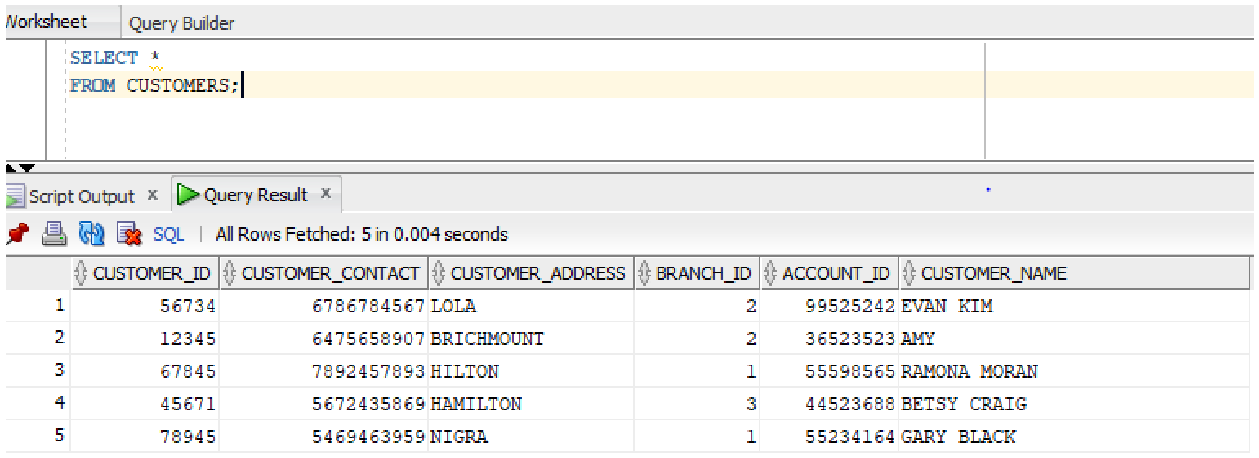


```
INSERT INTO CUSTOMERS(CUSTOMER_ID,CUSTOMER_CONTACT,CUSTOMER_ADDRESS,BRANCH_ID,ACCOUNT_ID,CUSTOMER_NAME)
VALUES(56734,6786784567,'LOLA',2,99525242,'EVAN KIM');

INSERT INTO CUSTOMERS(CUSTOMER_ID,CUSTOMER_CONTACT,CUSTOMER_ADDRESS,BRANCH_ID,ACCOUNT_ID,CUSTOMER_NAME)
VALUES(67845,7892457893,'HILTON',1,55598565,'RAMONA MORAN');

INSERT INTO CUSTOMERS(CUSTOMER_ID,CUSTOMER_CONTACT,CUSTOMER_ADDRESS,BRANCH_ID,ACCOUNT_ID,CUSTOMER_NAME)
VALUES(45671,5672435869,'HAMILTON',3,44523688,'BETSY CRAIG');

INSERT INTO CUSTOMERS(CUSTOMER_ID,CUSTOMER_CONTACT,CUSTOMER_ADDRESS,BRANCH_ID,ACCOUNT_ID,CUSTOMER_NAME)
VALUES(78945,5469463959,'NIGRA',1,55234164,'GARY BLACK');
```

Task completed in 0.015 seconds

```
1 row inserted.

1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

PRINT CUSTOMER TABLE



```
SELECT *
FROM CUSTOMERS;
```

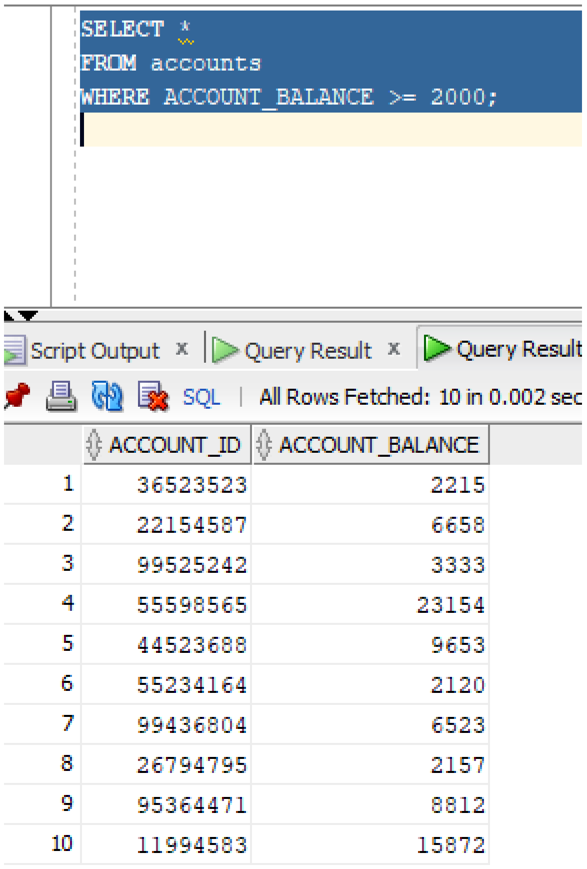
All Rows Fetched: 5 in 0.004 seconds

	CUSTOMER_ID	CUSTOMER_CONTACT	CUSTOMER_ADDRESS	BRANCH_ID	ACCOUNT_ID	CUSTOMER_NAME
1	56734	6786784567	LOLA	2	99525242	EVAN KIM
2	12345	6475658907	BRICHMOUNT	2	36523523	AMY
3	67845	7892457893	HILTON	1	55598565	RAMONA MORAN
4	45671	5672435869	HAMILTON	3	44523688	BETSY CRAIG
5	78945	5469463959	NIGRA	1	55234164	GARY BLACK

QUERIES:

1) List all accounts with balance above or equal 2000

```
SELECT *
FROM accounts
WHERE ACCOUNT_BALANCE >= 2000;
```



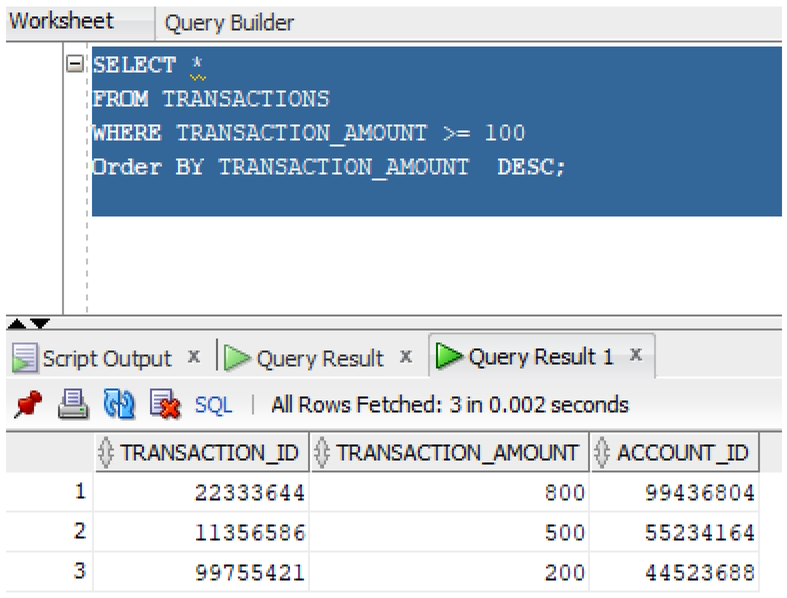
```
SELECT *
FROM accounts
WHERE ACCOUNT_BALANCE >= 2000;
```

	ACCOUNT_ID	ACCOUNT_BALANCE
1	36523523	2215
2	22154587	6658
3	99525242	3333
4	55598565	23154
5	44523688	9653
6	55234164	2120
7	99436804	6523
8	26794795	2157
9	95364471	8812
10	11994583	15872

Relational Algebra: σ ACCOUNT_BALANCE ≥2000 (ACCOUNTS)

2)List all transaction amounts greater than 100 in descending order.

SELECT *
FROM TRANSACTIONS
WHERE TRANSACTION_AMOUNT >= 100
Order BY TRANSACTION_AMOUNT DESC;



```
SELECT *
FROM TRANSACTIONS
WHERE TRANSACTION_AMOUNT >= 100
Order BY TRANSACTION_AMOUNT  DESC;
```

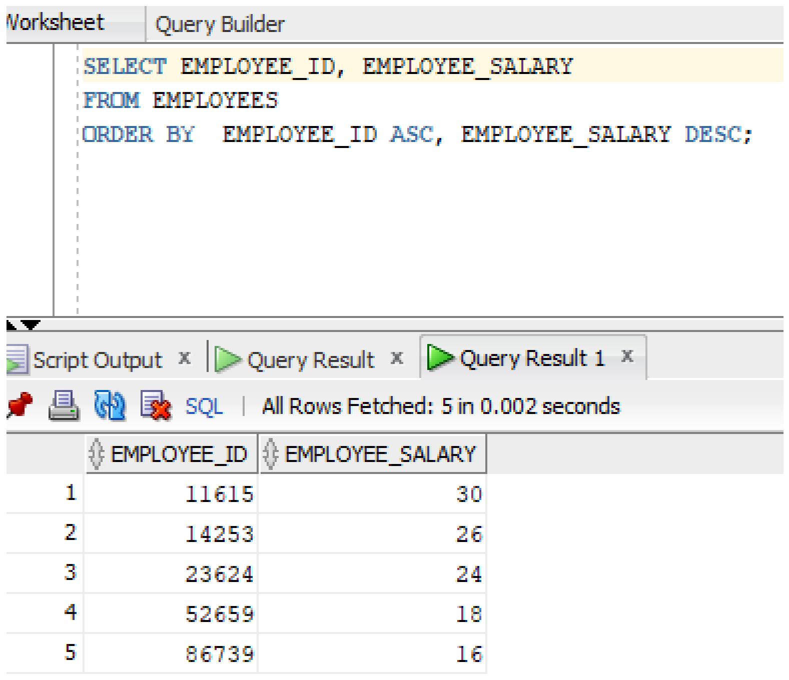
All Rows Fetched: 3 in 0.002 seconds

	TRANSACTION_ID	TRANSACTION_AMOUNT	ACCOUNT_ID
1	22333644	800	99436804
2	11356586	500	55234164
3	99755421	200	44523688

Relational Algebra: σ TRANSACTION_AMOUNT $\geq$ 100 (TRANSACTIONS)

3)List all employees(with employee ID in ascending order and salary in descending order.

SELECT EMPLOYEE_ID, EMPLOYEE_SALARY
FROM EMPLOYEES
ORDER BY EMPLOYEE_ID ASC, EMPLOYEE_SALARY DESC;



```
SELECT EMPLOYEE_ID, EMPLOYEE_SALARY
FROM EMPLOYEES
ORDER BY  EMPLOYEE_ID ASC, EMPLOYEE_SALARY DESC;
```

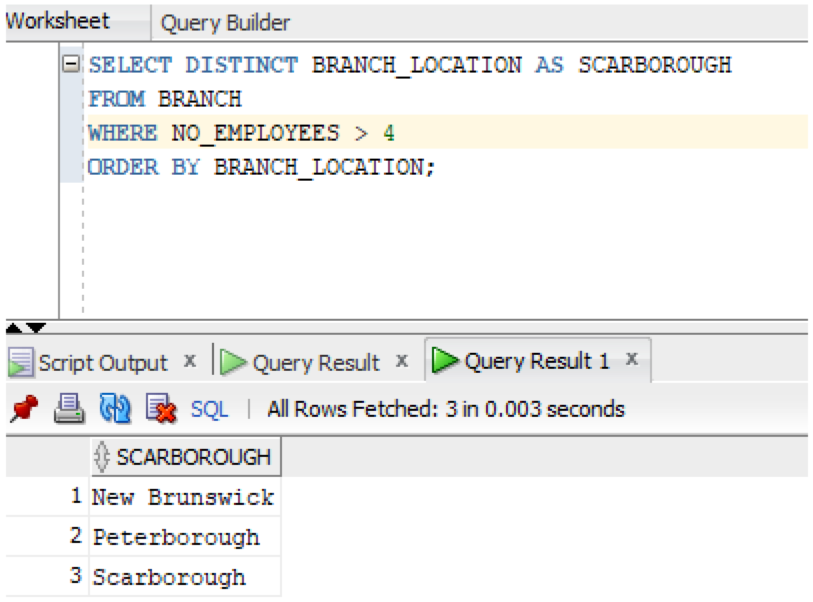
All Rows Fetched: 5 in 0.002 seconds

	EMPLOYEE_ID	EMPLOYEE_SALARY
1	11615	30
2	14253	26
3	23624	24
4	52659	18
5	86739	16

Relational Algebra: ΠEMPLOYEE_ID, EMPLOYEE_SALARY (EMPLOYEES)

4) List all areas with number of employees in bank greater than 4 and avoid printing branches with same location .

SELECT DISTINCT BRANCH_LOCATION AS SCARBOROUGH
FROM BRANCH
WHERE NO_EMPLOYEES > 4
ORDER BY BRANCH_LOCATION;



```
SELECT DISTINCT BRANCH_LOCATION AS SCARBOROUGH
FROM BRANCH
WHERE NO_EMPLOYEES > 4
ORDER BY BRANCH_LOCATION;
```

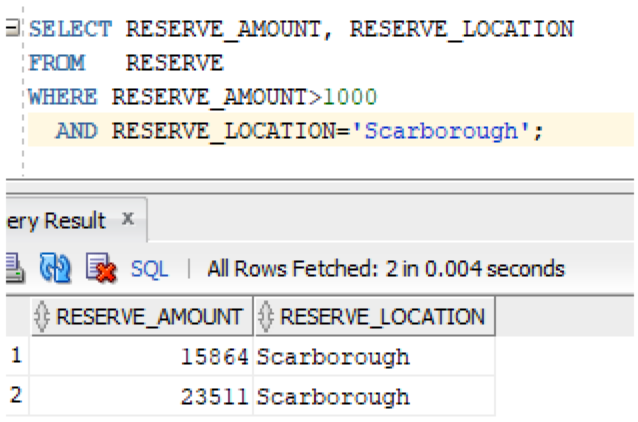
All Rows Fetched: 3 in 0.003 seconds

	SCARBOROUGH
1	New Brunswick
2	Peterborough
3	Scarborough

Relational Algebra: ΠBRANCH_LOCATION(σ NO_EMPLOYEE > 4 (BRANCH))

6) List all reserves (reserve amount and location only) greater than thousand and is in Scarborough area.

SELECT RESERVE_AMOUNT, RESERVE_LOCATION
FROM RESERVE
WHERE RESERVE_AMOUNT>1000
AND RESERVE_LOCATION='Scarborough';


```
SELECT RESERVE_AMOUNT, RESERVE_LOCATION
FROM   RESERVE
WHERE RESERVE_AMOUNT>1000
  AND RESERVE_LOCATION='Scarborough';
```

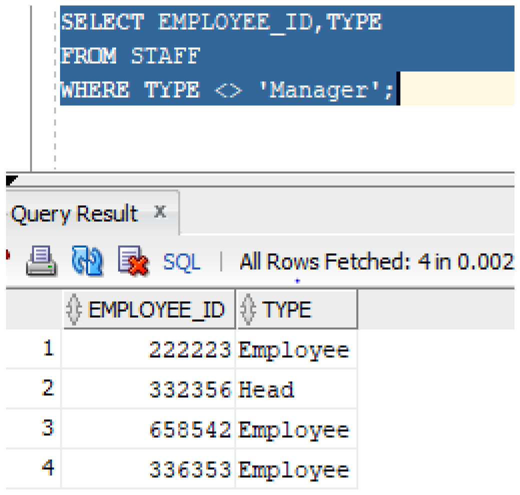
ery Result

SQL | All Rows Fetched: 2 in 0.004 seconds

	RESERVE_AMOUNT	RESERVE_LOCATION
1	15864	Scarborough
2	23511	Scarborough

Relational Algebra: ΠRESERVE_AMOUNT, RESERVE_LOCATION(σ RESERVE_AMOUNT > 1000 (RESERVE))

7) Find all types of staff in bank that the 'Manager'
SELECT EMPLOYEE_ID,TYPE
FROM STAFF
WHERE TYPE <> 'Manager';


```
SELECT EMPLOYEE_ID,TYPE
FROM STAFF
WHERE TYPE <> 'Manager';
```

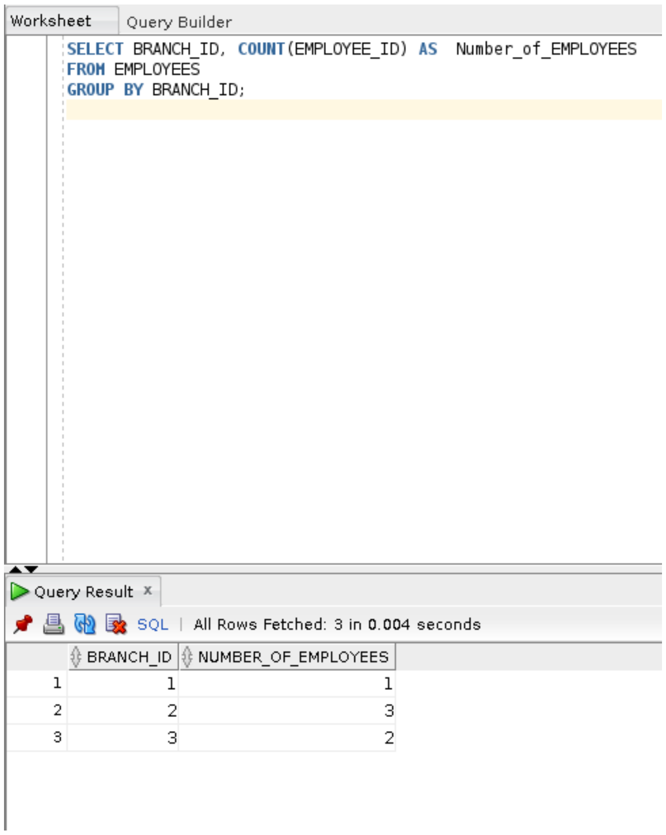
Query Result

SQL | All Rows Fetched: 4 in 0.002

	EMPLOYEE_ID	TYPE
1	222223	Employee
2	332356	Head
3	658542	Employee
4	336353	Employee

Relational Algebra: σ TYPE $\neq$'MANAGER' (STAFF)

8) Find the number of employees working in each branch.
SELECT BRANCH_ID, COUNT(EMPLOYEE_ID) AS Number_of_EMPLOYEES
FROM EMPLOYEES
GROUP BY BRANCH_ID;



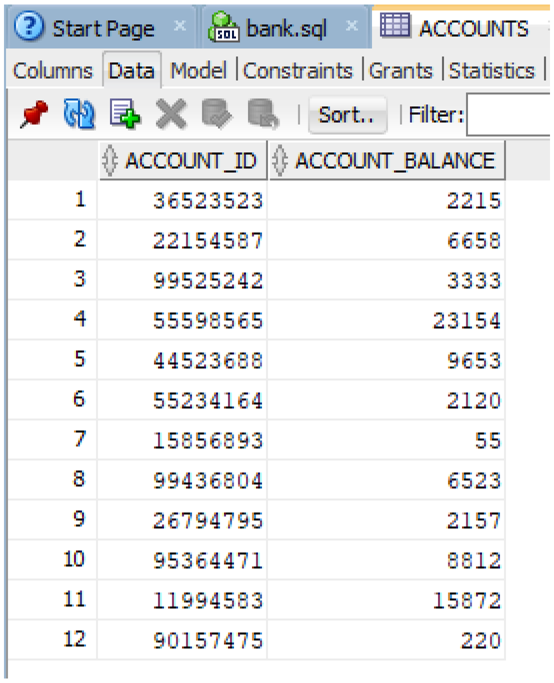
```
SELECT BRANCH_ID, COUNT(EMPLOYEE_ID) AS  Number_of_EMPLOYEES
FROM EMPLOYEES
GROUP BY BRANCH_ID;
```

```
All Rows Fetched: 3 in 0.004 seconds

     BRANCH_ID   NUMBER_OF_EMPLOYEES
1            1                     1
2            2                     3
3            3                     2
```

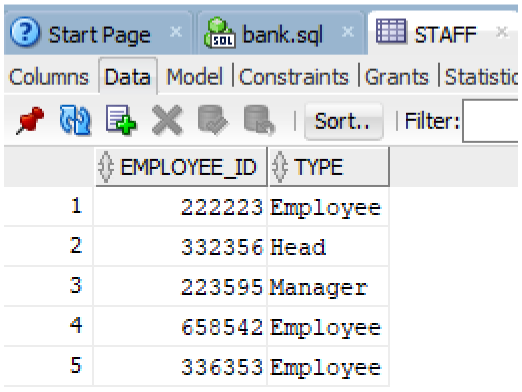
Relational Algebra: ΠBRANCH_ID (σ COUNT(EMPLOYEE_ID) (EMPLOYEE))

TABLES:

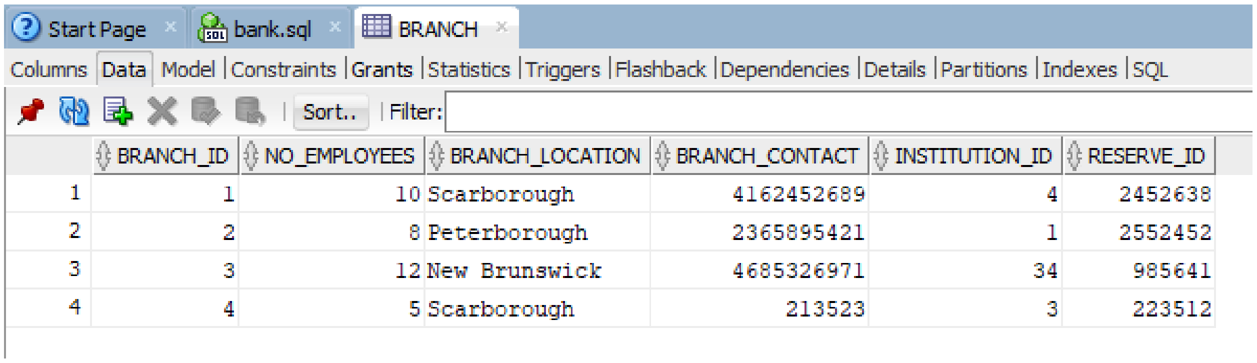
SELECT *
FROM CUSTOMERS;



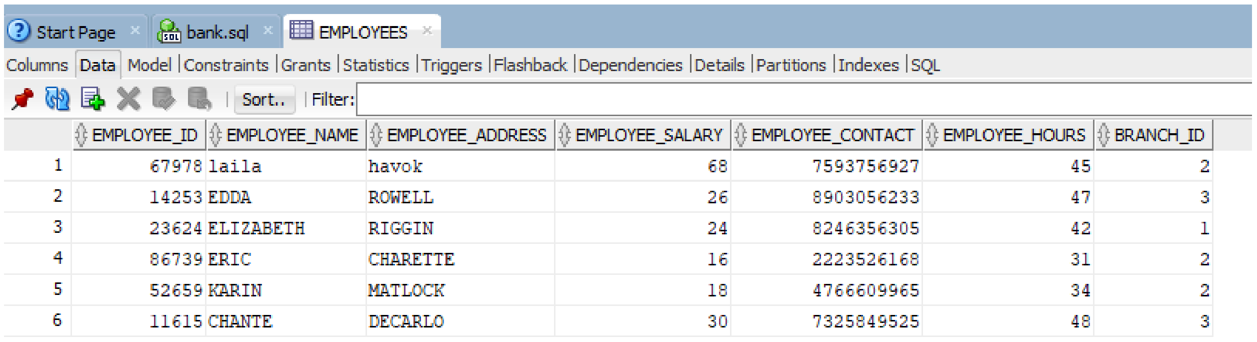
	ACCOUNT_ID	ACCOUNT_BALANCE
1	36523523	2215
2	22154587	6658
3	99525242	3333
4	55598565	23154
5	44523688	9653
6	55234164	2120
7	15856893	55
8	99436804	6523
9	26794795	2157
10	95364471	8812
11	11994583	15872
12	90157475	220



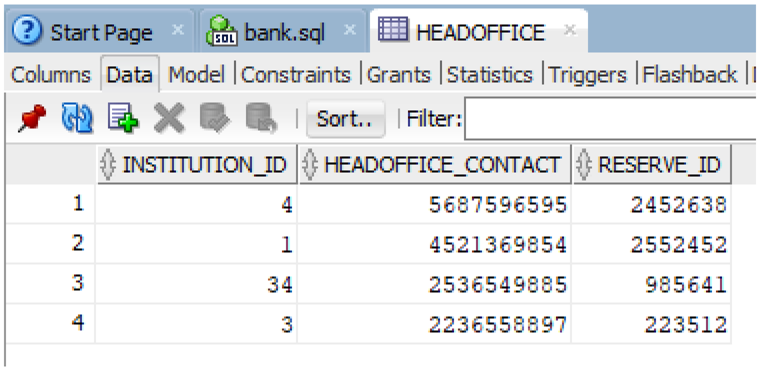
	EMPLOYEE_ID	TYPE
1	222223	Employee
2	332356	Head
3	223595	Manager
4	658542	Employee
5	336353	Employee


Start Page
bank.sql
BRANCH
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:

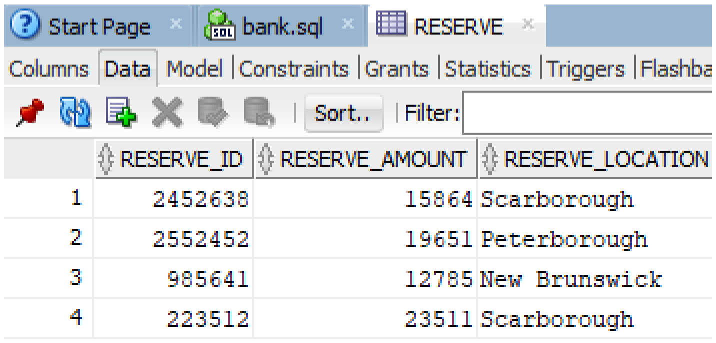
	BRANCH_ID	NO_EMPLOYEES	BRANCH_LOCATION	BRANCH_CONTACT	INSTITUTION_ID	RESERVE_ID
1	1	10	Scarborough	4162452689	4	2452638
2	2	8	Peterborough	2365895421	1	2552452
3	3	12	New Brunswick	4685326971	34	985641
4	4	5	Scarborough	213523	3	223512


Start Page
bank.sql
EMPLOYEES
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:

	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_ADDRESS	EMPLOYEE_SALARY	EMPLOYEE_CONTACT	EMPLOYEE_HOURS	BRANCH_ID
1	67978	laila	havok	68	7593756927	45	2
2	14253	EDDA	ROWELL	26	8903056233	47	3
3	23624	ELIZABETH	RIGGIN	24	8246356305	42	1
4	86739	ERIC	CHARETTE	16	2223526168	31	2
5	52659	KARIN	MATLOCK	18	4766609965	34	2
6	11615	CHANTE	DECARLO	30	7325849525	48	3

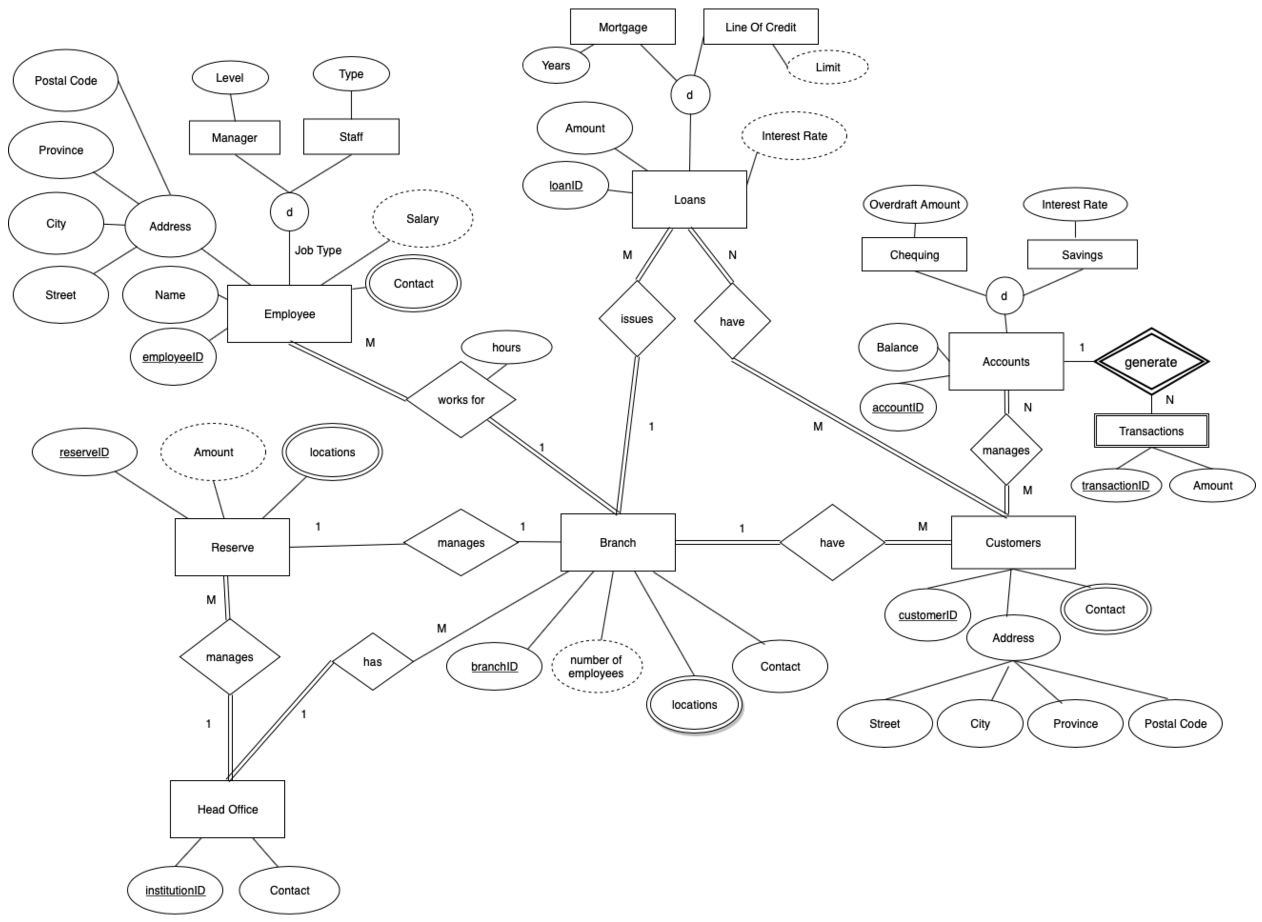


	INSTITUTION_ID	HEADOFFICE_CONTACT	RESERVE_ID
1	4	5687596595	2452638
2	1	4521369854	2552452
3	34	2536549885	985641
4	3	2236558897	223512



	RESERVE_ID	RESERVE_AMOUNT	RESERVE_LOCATION
1	2452638	15864	Scarborough
2	2552452	19651	Peterborough
3	985641	12785	New Brunswick
4	223512	23511	Scarborough

ER Diagram:



Mortgage
Line Of Credit
Years
Limit
d
Amount
Interest Rate
loanID
Loans
Postal Code
Level
Type
Province
Manager
Staff
City
Address
d
Salary
Job Type
Street
Name
Employee
Contact
employeeID
M
M
N
issues
have
Overdraft Amount
Interest Rate
Chequing
Savings
d
Balance
Accounts
1
generate
N
accountID
N
Transactions
manages
transactionID
Amount
M
hours
works for
1
1
M
reserveID
Amount
locations
Reserve
1
manages
1
Branch
1
have
M
Customers
M
manages
M
has
customerID
Contact
Address
1
1
branchID
number of employees
locations
Contact
Street
City
Province
Postal Code
Head Office
institutionID
Contact

Relational Schema:

BRANCH (branchID, institutionID, reserveID, numberOfEmployees, locations, contact)

-

EMPLOYEES (employeeID, branchID, name, salary, contact, hours)

employeeID → branch ID

employee ID → name
name → Employee ID

Salary → Employee ID

Employee ID → contact
contact → Employee ID

Employee ID → hours

MANAGER (employeeID, level)

STAFF(employeeID, type)

LOANS (loanID, branchID, customerID, amount, interestRate)

MORTGAGE (loanID, years)

LINEOFCREDIT (loadID, limit)

RESERVE (reserveID, amount, locations)

HEADOFFICE (institutionID, reserveID, contact)

CUSTOMERS (customerID, branchID, accountID, name, address, contact)

ACCOUNTS (accountID, balance)

CHEQUING (accountID, overDraftAmount)

SAVINGS (accountID, interestRate)

TRANSACTIONS (accountID, transactionID, amount)
ADDRESS (Street, City, Province, PostalCode)

SQL Source Code: Creating Tables

```
CREATE TABLE reserve(
  reserve_id NUMBER PRIMARY KEY,
  reserve_amount NUMBER NOT NULL,
  reserve_location VARCHAR2(50 CHAR)
);

CREATE TABLE headOffice(
  institution_id NUMBER PRIMARY KEY,
  headOffice_contact NUMBER DEFAULT 4169795000
);

CREATE TABLE branch(
  branch_id NUMBER PRIMARY KEY,
  institution_id NUMBER NOT NULL,
  reserve_id NUMBER NOT NULL,
  No_employees NUMBER CHECK (No_employees BETWEEN 1 AND 1000),
  branch_location VARCHAR2(50 CHAR),
  branch_contact  NUMBER UNIQUE,
  FOREIGN KEY (institution_id) REFERENCES headOffice(institution_id),
  FOREIGN KEY (reserve_id) REFERENCES reserve(reserve_id)
);

CREATE TABLE employees(
   employee_id NUMBER PRIMARY KEY ,
   branch_id NUMBER NOT NULL,
   employee_name VARCHAR2(50 CHAR) NOT NULL UNIQUE,
   employee_salary NUMBER NOT NULL,
   employee_contact NUMBER UNIQUE,
   employee_hours  NUMBER NOT NULL,
   FOREIGN KEY (branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
);

CREATE TABLE managers(
  employee_id NUMBER PRIMARY KEY ,
  employee_level NUMBER CHECK (employee_level BETWEEN 1 AND 5)
);
```

```
CREATE TABLE staff(
  employee_id NUMBER PRIMARY KEY ,
  employee_dept VARCHAR2(20 CHAR)
);

CREATE TABLE accounts(
    account_id NUMBER PRIMARY KEY ,
    account_balance NUMBER NOT NULL
);

CREATE TABLE chequing(
     account_id NUMBER PRIMARY KEY ,
     overdraft_amount NUMBER NOT NULL
);

CREATE TABLE savings(
     account_id NUMBER PRIMARY KEY,
     savings_interestRate NUMBER NOT NULL
);

CREATE TABLE customers(
      customer_id NUMBER PRIMARY KEY,
      branch_id NUMBER NOT NULL,
      account_id NUMBER NOT NULL,
      customer_name VARCHAR2(50 CHAR) NOT NULL,
      customer_contact NUMBER UNIQUE,
      FOREIGN KEY (branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE,
      FOREIGN KEY (account_id) REFERENCES accounts(account_id) ON DELETE CASCADE
);

CREATE TABLE loans(
      loan_id NUMBER PRIMARY KEY,
      branch_id NUMBER NOT NULL,
      customer_id NUMBER NOT NULL,
      loan_amount NUMBER NOT NULL,
      loan_interestRate NUMBER NOT NULL,
      FOREIGN KEY (branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE,
      FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE
CASCADE
);
```

```
CREATE TABLE mortgage(
   loan_id NUMBER PRIMARY KEY,
   mortgage_years NUMBER CHECK (mortgage_years BETWEEN 1 AND 30)
);

CREATE TABLE lineOfCredit(
   loan_id NUMBER PRIMARY KEY,
   loc_limit NUMBER CHECK (loc_limit BETWEEN 1 AND 10000)
);


CREATE TABLE transactions(
     account_id NUMBER NOT NULL,
     transaction_id NUMBER NOT NULL UNIQUE,
     transaction_amount NUMBER NOT NULL,
     CONSTRAINT transactions_pk PRIMARY KEY(account_id, transaction_id),
     FOREIGN KEY (account_id) REFERENCES accounts(account_id) ON DELETE CASCADE
);


CREATE TABLE address(
  customer_id NUMBER UNIQUE,
  employee_id NUMBER UNIQUE,
  street VARCHAR2(50 CHAR),
  city VARCHAR2(50 CHAR),
  province VARCHAR2(50 CHAR),
  postal_code VARCHAR2(50 CHAR)
);
```

<u>Inserting Data into Tables: (Sample)</u>

```
SELECT * FROM BRANCH;
SELECT * FROM RESERVE;
SELECT * FROM EMPLOYEES;
UPDATE EMPLOYEES SET EMPLOYEE_CONTACT=7059872346 WHERE EMPLOYEE_ID =
3004;

INSERT ALL
  INTO
EMPLOYEES(EMPLOYEE_ID,BRANCH_ID,EMPLOYEE_NAME,EMPLOYEE_SALARY,EMPL
OYEE_CONTACT,EMPLOYEE_HOURS)
  VALUES(3001,34553,'Neil Caruthers',45334,7053839293,32)
  INTO
EMPLOYEES(EMPLOYEE_ID,BRANCH_ID,EMPLOYEE_NAME,EMPLOYEE_SALARY,EMPL
OYEE_CONTACT,EMPLOYEE_HOURS)
  VALUES(3002,34553,'Latrice Seely',57842,705834568,39)
  INTO
EMPLOYEES(EMPLOYEE_ID,BRANCH_ID,EMPLOYEE_NAME,EMPLOYEE_SALARY,EMPL
OYEE_CONTACT,EMPLOYEE_HOURS)
  VALUES(3003,34553,'Leia Mickle',68844,7053839877,46)
  INTO
EMPLOYEES(EMPLOYEE_ID,BRANCH_ID,EMPLOYEE_NAME,EMPLOYEE_SALARY,EMPL
OYEE_CONTACT,EMPLOYEE_HOURS)
  VALUES(3004,34553,'Valerie Culp',76844,705832568,27)
  INTO
EMPLOYEES(EMPLOYEE_ID,BRANCH_ID,EMPLOYEE_NAME,EMPLOYEE_SALARY,EMPL
OYEE_CONTACT,EMPLOYEE_HOURS)
  VALUES(3005,34553,'Kazuko Engles',69442,7053939877,25)
SELECT * FROM dual;
```