1)

Code:

```
CREATE DATABASE w4q1_Bonk;
USE w4q1_Bonk;

CREATE TABLE branch (
  branch_name VARCHAR(255) PRIMARY KEY,
  branch_city VARCHAR (255) NOT NULL,
  assets INT NOT NULL
);
INSERT INTO branch (branch_name, branch_city, assets) VALUES
            ("Brighton"  , "Brooklyn"  , 71E6),
            ("Downtown"  , "Brooklyn"  , 90E5),
            ("Mianus"    , "Horseneck" ,  4E5),
            ("North Town", "Rye"       , 37E5),
            ("Perryridge", "Horseneck" , 17E5),
            ("Pownal"    , "Bennington",  3E5),
            ("Redwood"   , "Palo Alto" , 21E5),
            ("Round Hill", "Horseneck" , 80E5);

CREATE TABLE customer (
  customer_name VARCHAR(255) PRIMARY KEY,
  customer_street VARCHAR(255) NOT NULL,
  customer_city VARCHAR(255) NOT NULL
);
INSERT INTO customer (customer_name, customer_street, customer_city) VALUES
            ("Adams"   , "Spring"   , "Pittsfield"),
            ("Brooks"  , "Senator"  , "Brooklyn"  ),
            ("Curry"   , "North"    , "Rye"       ),
            ("Glenn"   , "Sand Hill", "Woodside"  ),
            ("Green"   , "Walnut"   , "Stamford"  ),
```

```
        ("Hayes"   , "Main"     , "Harrison"  ),
        ("Johnson" , "Alma"     , "Palo Alto" ),
        ("Jones"   , "Main"     , "Harrison"  ),
        ("Lindsay" , "Park"     , "Pittsfield"),
        ("Smith"   , "North"    , "Rye"       ),
        ("Turner"  , "Putnam"   , "Stamford"  ),
        ("Jackson" , "East"     , "Downtown"  ),
   ("Williams", "West"     , "Downtown"  );

-- Seems like `account` can be used as variable
-- and is not actually a keyword
CREATE TABLE accounts (
  account_number VARCHAR(255) PRIMARY KEY,
  branch_name VARCHAR(255) NOT NULL,
  balance INT NOT NULL,
  FOREIGN KEY (branch_name) REFERENCES branch(branch_name),
  CHECK (account_number REGEXP '^A\-[1-9][0-9]*$')
);
INSERT INTO accounts (account_number, branch_name, balance) VALUES
        ("A-101", "Downtown"  , 500),
        ("A-102", "Perryridge", 400),
        ("A-201", "Brighton"  , 900),
        ("A-215", "Mianus"    , 700),
        ("A-217", "Brighton"  , 750),
        ("A-222", "Redwood"   , 700),
        ("A-305", "Round Hill", 350);

CREATE TABLE loan (
  loan_number VARCHAR(255) PRIMARY KEY,
  branch_name VARCHAR(255) NOT NULL,
  amount INT NOT NULL,
```

```
  FOREIGN KEY (branch_name) REFERENCES branch(branch_name),
  CHECK (loan_number REGEXP '^L\-\[1-9][0-9]*$')
);
INSERT INTO loan (loan_number, branch_name, amount) VALUES
        ("L-11", "Round Hill",  900),
        ("L-14", "Downtown"  , 1500),
        ("L-15", "Perryridge", 1500),
        ("L-16", "Perryridge", 1300),
        ("L-17", "Downtown"  , 1000),
        ("L-23", "Redwood"   , 2000),
        ("L-93", "Mianus"    ,  500);

CREATE TABLE depositor(
  customer_name VARCHAR(255) NOT NULL,
  account_number VARCHAR(255) NOT NULL,
  FOREIGN KEY (customer_name) REFERENCES customer(customer_name),
  FOREIGN KEY (account_number) REFERENCES accounts(account_number)
);
INSERT INTO depositor (customer_name, account_number) VALUES
        ("Hayes"  , "A-102"),
        ("Jackson", "A-101"),
        ("Johnson", "A-201"),
        ("Jones"  , "A-217"),
        ("Lindsay", "A-222"),
        ("Smith"  , "A-215"),
        ("Turner" , "A-305");

CREATE TABLE borrower(
  loan_number VARCHAR(255) NOT NULL,
  customer_name VARCHAR(255) NOT NULL,
  FOREIGN KEY (loan_number) REFERENCES loan(loan_number),
```

```
  FOREIGN KEY (customer_name) REFERENCES customer(customer_name)
);
INSERT INTO borrower (customer_name, loan_number) VALUES
        ("Adams"   , "L-16"),
        ("Curry"   , "L-93"),
        ("Hayes"   , "L-15"),
        ("Jackson" , "L-14"),
        ("Jones"   , "L-17"),
        ("Smith"   , "L-11"),
        ("Smith"   , "L-23"),
        ("Williams", "L-17");



DELETE FROM customers;
DROP TABLE depositor;
DROP DATABASE w4q1_bonk;
```

Inputs:

- ➔ Branch Table:

branch_name	branch_city	assets
Brighton	Brooklyn	71000000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000
NULL	NULL	NULL

- ➔ Customer Table:

	customer_name	customer_street	customer_city
▶	Adams	Spring	Pittsfield
	Brooks	Senator	Brooklyn
	Curry	North	Rye
	Glenn	Sand Hill	Woodside
	Green	Walnut	Stamford
	Hayes	Main	Harrison
	Jackson	East	Downtown
	Johnson	Alma	Palo Alto
	Jones	Main	Harrison
	Lindsay	Park	Pittsfield
	Smith	North	Rye
	Turner	Putnam	Stamford
	Williams	West	Downtown
*	NULL	NULL	NULL

➔ Accounts Table:

	account_number	branch_name	balance
▶	A-101	Downtown	500
	A-102	Perryridge	400
	A-201	Brighton	900
	A-215	Mianus	700
	A-217	Brighton	750
	A-222	Redwood	700
	A-305	Round Hill	350
*	NULL	NULL	NULL

➔ Loan Table:

	branch_name	customer_name
▶	Perryridge	Hayes
	Downtown	Jackson
	Brighton	Johnson
	Brighton	Jones
	Redwood	Lindsay
	Mianus	Smith
	Round Hill	Turner
	Perryridge	Adams
	Mianus	Curry
	Downtown	Jones
	Round Hill	Smith
	Redwood	Smith
	Downtown	Williams

➔ Depositor Relation:

	customer_name	account_number
▶	Hayes	A-102
	Jackson	A-101
	Johnson	A-201
	Jones	A-217
	Lindsay	A-222
	Smith	A-215
	Turner	A-305

➔ Borrower Relation:

	branch_name	customer_name
▶	Perryridge	Hayes
	Downtown	Jackson
	Brighton	Johnson
	Brighton	Jones
	Redwood	Lindsay
	Mianus	Smith
	Round Hill	Turner
	Perryridge	Adams
	Mianus	Curry
	Downtown	Jones
	Round Hill	Smith
	Redwood	Smith
	Downtown	Williams

Queries & Outputs:

```
-- 1a) Create a view consisting of branch names and the names of customers who have either
-- an account or a loan at that branch. Assume that view to be called all-customer.
CREATE VIEW all_customer AS
SELECT branch_name, customer_name
FROM depositor, accounts
WHERE depositor.account_number = accounts.account_number
UNION
(SELECT branch_name , customer_name
FROM borrower , loan
```

WHERE borrower.loan_number = loan.loan_number);

-- For Output

SELECT * FROM all_customer;

branch_name	customer_name
Perryridge	Hayes
Downtown	Jackson
Brighton	Johnson
Brighton	Jones
Redwood	Lindsay
Mianus	Smith
Round Hill	Turner
Perryridge	Adams
Mianus	Curry
Downtown	Jones
Round Hill	Smith
Redwood	Smith
Downtown	Williams

-- 1b) Create a view gives for each branch the sum of the amounts of all the loans at the branch

CREATE VIEW sum_of_loans AS

SELECT branch_name , SUM(amount)

FROM loan

GROUP BY branch_name;

-- For Output

SELECT * FROM sum_Of_Loans;

branch_name	SUM(amount)
Downtown	2500
Mianus	500
Perryridge	2800
Redwood	2000
Round Hill	900

-- 1c) Using the view all-customer, we can find all customersof the Perryridgebranch

```
SELECT customer_name
FROM all_customer
WHERE branch_name = "Perryridge";
```

customer_name
Hayes
Adams

-- 1d Write a Query for below Relational algebraic notation :

```
SELECT depositor.customer_name
FROM depositor
UNION
(SELECT  borrower.customer_name
FROM borrower);
```

customer_name
Hayes
Jackson
Johnson
Jones
Lindsay
Smith
Turner
Adams
Curry
Williams

2)

Code:

```
CREATE DATABASE Sasalele;
USE sasalele;
```

```
CREATE TABLE Salesman (
  salesman_id INT PRIMARY KEY,
  sname VARCHAR(25) NOT NULL,
  city VARCHAR(21) NOT NULL,
  commission FLOAT(3,2) NOT NULL
);
INSERT INTO Salesman VALUES
  (5001, 'James Hoog', 'New York', 0.15),
  (5002, 'Nail Knite', 'Paris'   , 0.13),
  (5005, 'Pit Alex'  , 'London'  , 0.11),
  (5006, 'Mc Lyon'   , 'Paris'   , 0.14),
  (5007, 'Paul Adam' , 'Rome'    , 0.13),
  (5003, 'Lauson Hen', 'San Jose', 0.12);

CREATE TABLE Customer (
  customer_id INT PRIMARY KEY,
  cust_name VARCHAR(25) NOT NULL,
  city VARCHAR(20) NOT NULL,
  grade INT NOT NULL,
  salesman_id INT,
  FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
);
INSERT INTO Customer VALUES
  (3002, 'Nick Rimando'  , 'New York' , 100, 5001),
  (3007, 'Brad Davis'    , 'New York' , 200, 5001),
  (3005, 'Graham Zusi'   , 'Califonia', 200, 5002),
  (3008, 'Julian Green'  , 'London'   , 300, 5002),
  (3004, 'Fabian Johnson', 'Paris'    , 300, 5006),
  (3009, 'Geoff Cameron' , 'Berlin'   , 100, 5003),
  (3003, 'Jozy Altidor'  , 'Moscow'   , 200, 5007),
  (3001, 'Brad Guzan'    , 'London'   , 300, 5005);
```

```
CREATE TABLE Orders (
  ord_no INT PRIMARY KEY,
  purch_amt FLOAT(6,2) NOT NULL,
  ord_date DATE NOT NULL,
  customer_id INT NOT NULL,
  FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
  salesman_id INT NOT NULL,
  FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
);
INSERT INTO Orders VALUES
        (70001,  150.5 , '2012-10-05', 3005, 5002),
        (70009,  270.65, '2012-09-10', 3001, 5005),
        (70002,   65.26, '2012-10-05', 3002, 5001),
        (70004,  110.5 , '2012-08-17', 3009, 5003),
        (70007,  948.5 , '2012-09-10', 3005, 5002),
        (70005, 2400.6 , '2012-07-27', 3007, 5001),
        (70008, 5760   , '2012-09-10', 3002, 5001),
        (70010, 1983.43, '2012-10-10', 3004, 5006),
        (70003, 2480.4 , '2012-10-10', 3009, 5003),
        (70012,  250.45, '2012-06-27', 3008, 5002),
        (70011,   75.29, '2012-08-17', 3003, 5007),
        (70013, 3045.6 , '2012-04-25', 3002, 5001);



DELETE FROM Orders;
DELETE FROM Customer;
DELETE FROM Salesman;
DROP TABLE Customer, Salesman, Orders;
DROP DATABASE Sasalele;
```

Inputs:

➔ Customer Table:

customer_id	cust_name	city	grade	salesman_id
3001	Brad Guzan	London	300	5005
3002	Nick Rimando	New York	100	5001
3003	Jozy Altidor	Moscow	200	5007
3004	Fabian Johnson	Paris	300	5006
3005	Graham Zusi	Califonia	200	5002
3007	Brad Davis	New York	200	5001
3008	Julian Green	London	300	5002
3009	Geoff Cameron	Berlin	100	5003
NULL	NULL	NULL	NULL	NULL

➔ Salesman Table:

salesman_id	sname	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
NULL	NULL	NULL	NULL

➔ Orders Relation:

ord_no	purch_amt	ord_date	customer_id	salesman_id
70003	2480.40	2012-10-10	3009	5003
70004	110.50	2012-08-17	3009	5003
70005	2400.60	2012-07-27	3007	5001
70007	948.50	2012-09-10	3005	5002
70008	5760.00	2012-09-10	3002	5001
70009	270.65	2012-09-10	3001	5005
70010	1983.43	2012-10-10	3004	5006
70011	75.29	2012-08-17	3003	5007
70012	250.45	2012-06-27	3008	5002
70013	3045.60	2012-04-25	3002	5001
NULL	NULL	NULL	NULL	NULL

Queries & Outputs:

-- 2a) From the table, create a view to count the number of customers in each grade.

CREATE VIEW grade_count (grade, count)

AS SELECT grade, COUNT(*)

FROM customer

GROUP BY grade;

SELECT * from grade_count;

	grade	count
▶	100	2
	200	3
	300	3

-- 2b) From the following table, create a view to count the number of unique

-- customer, compute average and total purchase amount of customer orders by

-- each date.

CREATE VIEW total_by_date

AS SELECT ord_date, COUNT(DISTINCT customer_id),

AVG(purch_amt), SUM(purch_amt)

FROM orders

GROUP BY ord_date;

SELECT * FROM total_by_date;

	ord_date	COUNT(DISTINCT customer_id)	AVG(purch_amt)	SUM(purch_amt)
▶	2012-04-25	1	3045.600098	3045.60
	2012-06-27	1	250.449997	250.45
	2012-07-27	1	2400.600098	2400.60
	2012-08-17	2	92.895000	185.79
	2012-09-10	3	2326.383331	6979.15
	2012-10-05	2	107.880001	215.76
	2012-10-10	2	2231.914978	4463.83

-- 2c) Create a view to get the salesperson and customer by name. Return order

-- name, purchase amount, salesperson ID, name, customer name.

```
CREATE VIEW order_details
AS SELECT ord_no, purch_amt, a.salesman_id, sname, cust_name
FROM orders a, customer b, salesman c
WHERE a.customer_id = b.customer_id
AND a.salesman_id = c.salesman_id;
```

```
SELECT * FROM order_details;
```

ord_no	purch_amt	salesman_id	sname	cust_name
70002	65.26	5001	James Hoog	Nick Rimando
70005	2400.60	5001	James Hoog	Brad Davis
70008	5760.00	5001	James Hoog	Nick Rimando
70013	3045.60	5001	James Hoog	Nick Rimando
70001	150.50	5002	Nail Knite	Graham Zusi
70007	948.50	5002	Nail Knite	Graham Zusi
70012	250.45	5002	Nail Knite	Julian Green
70003	2480.40	5003	Lauson Hen	Geoff Cameron
70004	110.50	5003	Lauson Hen	Geoff Cameron
70009	270.65	5005	Pit Alex	Brad Guzan
70010	1983.43	5006	Mc Lyon	Fabian Johnson
70011	75.29	5007	Paul Adam	Jozy Altidor

-- 2d) Create a view to find the salespersons who issued orders on October 10th,

-- 2012. Return all the fields of salesperson.

```
CREATE VIEW salesman_oct_10
AS SELECT *
FROM salesman
WHERE salesman_id IN
   (SELECT salesman_id
       FROM orders
       WHERE ord_date = '2012-10-10');
```

```
SELECT * FROM salesman_oct_10;
```

	salesman_id	sname	city	commission
▶	5003	Lauson Hen	San Jose	0.12
	5006	Mc Lyon	Paris	0.14