Database Management System (DBMS)

Lecture-25

Dharmendra Kumar

October 19, 2020

Set Operations

The SQL operations union, intersect, and except operate on relations and correspond to the relational algebra operations $\cup$, $\cap$, and -.

1. Find all customers having a loan, an account, or both at the bank.
2. Find all customers who have both a loan and an account at the bank.
3. Find all customers who have an account but no loan at the bank.

SQL

Solution:

1. (select customer-name
   from depositor)
   union
   (select customer-name
   from borrower)
2. (select customer-name
   from depositor)
   intersect
   (select customer-name
   from borrower)
3. (select customer-name
   from depositor)
   except
   (select customer-name
   from borrower)

Aggregate Functions

Aggregate functions are functions that take a collection (a set or multiset) of values as input and return a single value. SQL offers five built-in aggregate functions:

1. Average: avg
2. Minimum: min
3. Maximum: max
4. Total: sum
5. Count: count

Example:

1. Find the average account balance at the Perryridge branch.
2. Find the average account balance at each branch.
3. Find the number of depositors for each branch.

Solution:

1. select avg (balance)
   from account
   where branch-name = 'Perryridge'
2. select branch-name, avg (balance)
   from account
   group by branch-name
3. select branch-name, count (distinct customer-name)
   from depositor, account
   where depositor.account-number = account.account-number
   group by branch-name

Example:

1. Find the branches where the average account balance is more than $1200.
2. Find the average balance for each customer who lives in Harrison and has at least three accounts.

Solution:

1. select branch-name, avg (balance)
   from account
   group by branch-name
   having avg (balance) $>$ 1200
2. select depositor.customer-name, avg (balance)
   from depositor, account, customer
   where depositor.account-number $=$ account.account-number
   and depositor.customer-name $=$ customer.customer-name and
   customer-city $=$ 'Harrison'
   group by depositor.customer-name
   having count (distinct depositor.account-number) ≥ 3

Nested Subqueries

1. Find all customers who have both a loan and an account at the bank.
2. Find all customers who have both an account and a loan at the Perryridge branch.

SQL

Solution:

1. select distinct customer-name
   from borrower
   where customer-name in (select customer-name from depositor)
2. select distinct customer-name
   from borrower, loan
   where borrower.loan-number = loan.loan-number and branch-name = 'Perryridge' and (branch-name, customer-name) in (select branch-name, customer-name
   from depositor, account
   where depositor.account-number = account.account-number)