Database Management System (DBMS)

Lecture-15

Dharmendra Kumar

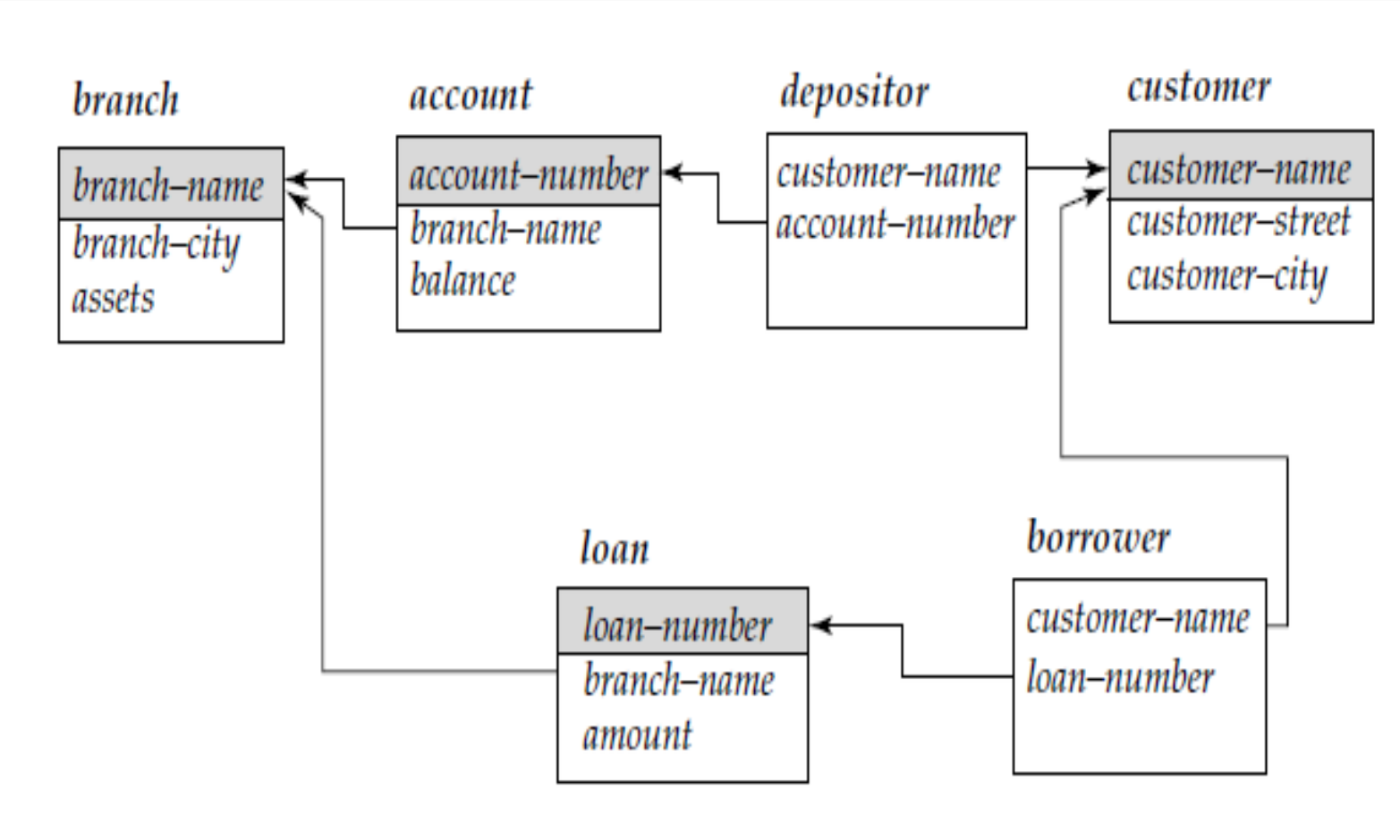
September 7, 2020

Schema Diagram

- A database schema, along with primary key and foreign key dependencies, can be depicted pictorially by schema diagrams.
- Each relation appears as a box, with the attributes listed inside it and the relation name above it. If there are primary key attributes, a horizontal line crosses the box, with the primary key attributes listed above the line. Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation.

Relational Model

Example: Following figure shows the schema diagram for our banking enterprise.



Query Languages

A query language is a language in which a user requests information from the database. Query languages can be categorized as either procedural or non-procedural.

In a procedural language, the user instructs the system to perform a sequence of operations on the database to compute the desired result.

In a non-procedural language, the user describes the desired information without giving a specific procedure for obtaining that information.

Relational Model

In this chapter, we will study following three languages:-

1. Relational algebra
2. Tuple relational calculus
3. Domain relational calculus

In these languages, relational algebra is a procedural but tuple and domain relational calculus are non-procedural languages.

Relational Model

Consider the following banking database. We will write all the queries for this database.

branch

branch-name	*branch-city*	*assets*
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

customer

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
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

account

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

depositor

customer-name	*account-number*
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

loan

loan-number	*branch-name*	*amount*
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

borrower

customer-name	*loan-number*
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Figure 1: Banking database

Relational Algebra

- The relational algebra is a procedural query language.
- It consists of a set of operations that take one or two relations as input and produce a new relation as their result.
- The fundamental operations in the relational algebra are select, project, union, set difference, Cartesian product, and rename. In addition to the fundamental operations, there are several other operations—namely, set intersection, natural join, division, and assignment.

Fundamental Operations

- The select, project, and rename operations are called unary operations, because they operate on one relation.
- The other three operations operate on pairs of relations and are, therefore, called binary operations.

The Select Operation

The select operation selects tuples that satisfy a given predicate. We use the lowercase Greek letter sigma (σ) to denote selection. The predicate appears as a subscript to σ. The argument relation is in parentheses after the σ. That is,

$$\sigma_P(r)$$

Here, r is a name of a relation and P is a predicate.

Example: Select those tuples of the loan relation where the branch is "Perryridge".

Relational Algebra

Solution: $\sigma_{branch-name=\text{"}Perryridge\text{"}}(loan)$

loan-number	*branch-name*	*amount*
L-15	Perryridge	1500
L-16	Perryridge	1300

Relational Algebra

Example: Find all tuples in which the amount lent is more than \$1200.

Solution: $\sigma_{amount>1200}(loan)$

Note: In general, we allow comparisons using $=, \neq, <, \leq, >, \geq$ in the selection predicate. Furthermore, we can combine several predicates into a larger predicate by using the connectives and ($\wedge$), or ($\vee$), and not ().

Example: Find those tuples pertaining to loans of more than \$1200 made by the "Perryridge" branch.

Solution: $\sigma_{(branch-name=\text{"}Perryridge\text{"})\wedge(amount>1200)}(loan)$

The Project Operation

The project operation is used to select columns of a table. It is denoted by Π. We list those attributes that we wish to appear in the result as a subscript to Π. The argument relation follows in parentheses.

$$\Pi_{A,B,C}(r)$$

Here, r is a name of a relation and A, B, C are the attributes corresponding selected column.

o.6cm

Example: List all loan numbers and the amount of the loan.

Relational Algebra

Solution: $\Pi_{loan-number,amount}(loan)$

loan-number	*amount*
L-11	900
L-14	1500
L-15	1500
L-16	1300
L-17	1000
L-23	2000
L-93	500

Example: Find those customers who live in Harrison.

Solution: $\Pi_{customer-name}(\sigma_{city="Harrison"}(Customer))$