RELATIONAL MODEL

Database Management System

SUJAN TAMRAKAR

Relational Model

- Most dominant data model for db
- Very simple and elegant model for data operation
- Allows for a very powerful query & manipulation languages
- Represents db as a collection of relations
- Relation = relvar = table
- Each relation is a table with rows & columns
- A row of a relation is called a tuple
- Column head is called field or attribute
- Domain is a set of atomic values, it indicates the valid set of values that it can take
- Eg. Domain of SGPA must be real (2.5, 3.4, 3.9, 4)
- A data type is specified for each domain (eg: age = int, name = text/string/varchar, etc.)

Relational Model

- Relational Schema ‘R’ is denoted by R(A1, A2, A3, … , An)

  where R = name of relation; A1, A2,… = list of attributes
- Relational Schema describes a relation
- Degree is the number of attributes of its relational schema.
- A relational state ‘r’ of relational schema R(A1, A2, A3, … , An) is denoted by r(R) and is a set of ‘n’ tuples

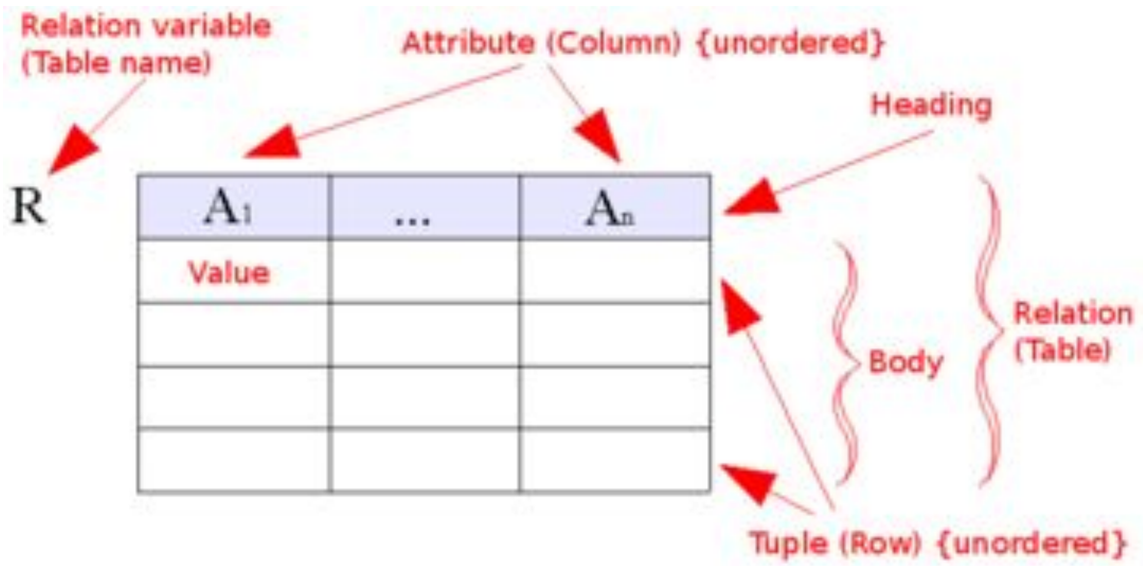
  i.e. r(R) = r = {t1, t2, t3, … , tn}
- Each ‘n’ tuple t is an ordered list of n values, t = <v1, v2, v3, … , vn>

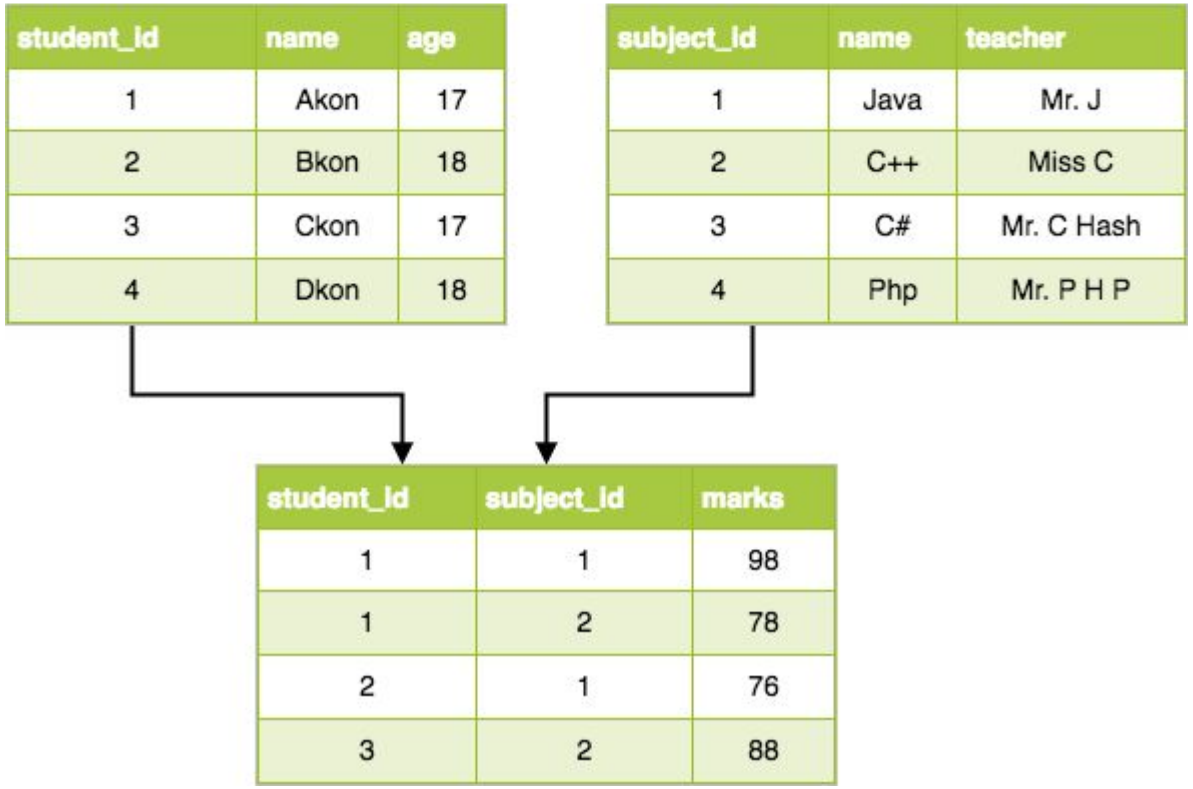
Relational Model Constraints

- Constraints are restrictions on actual values in db state
- Types:

a) Domain constraint

b) Key constraint



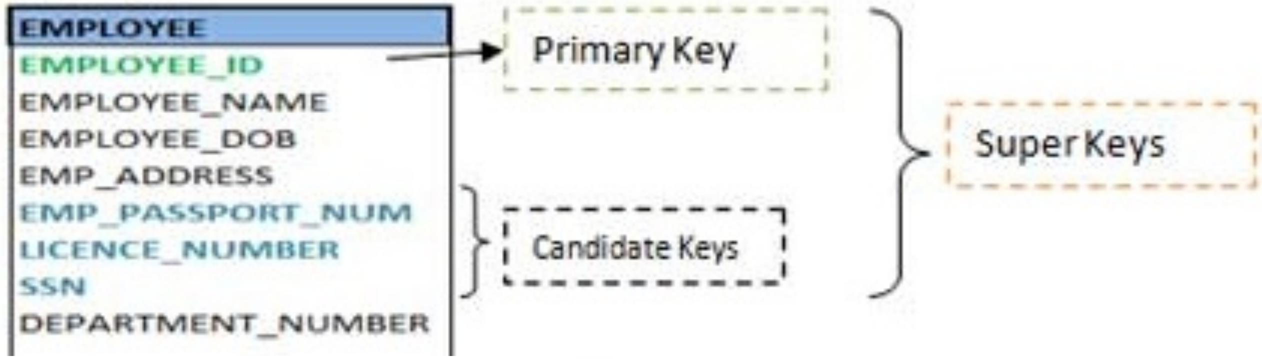


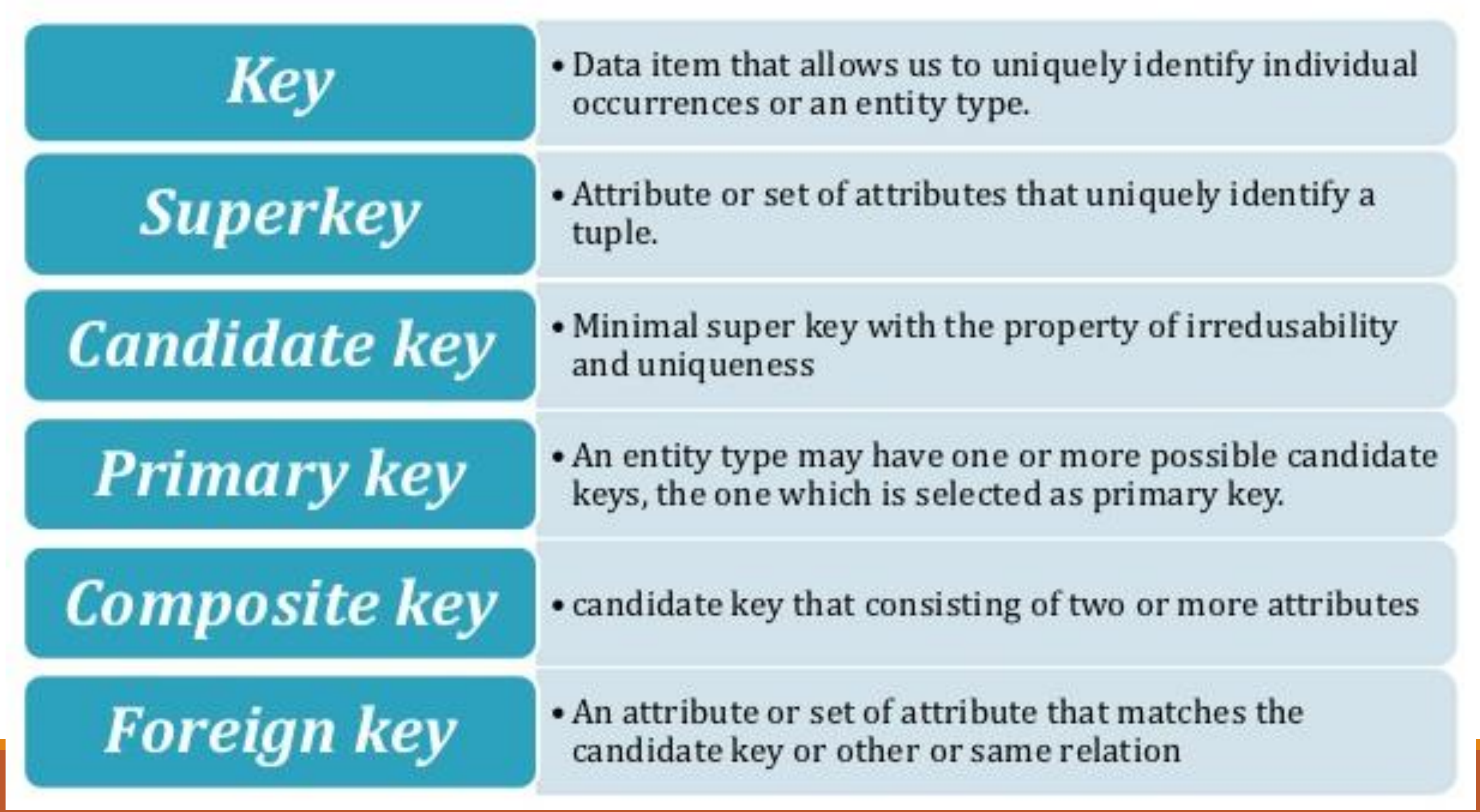
student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

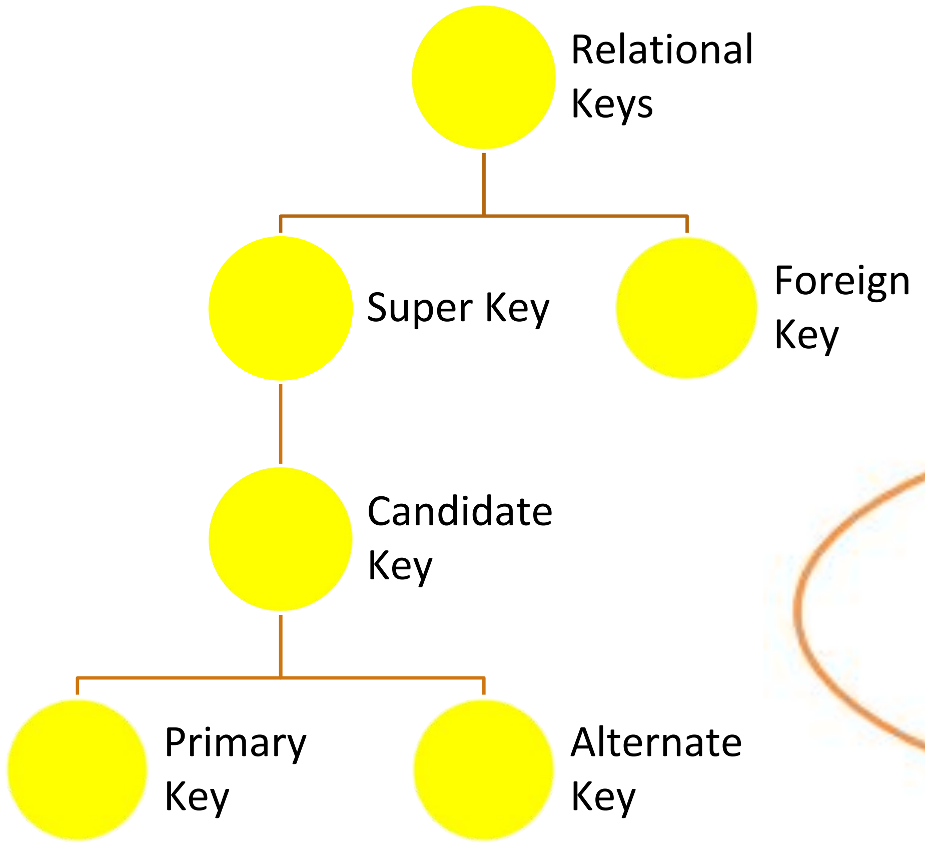
Keys

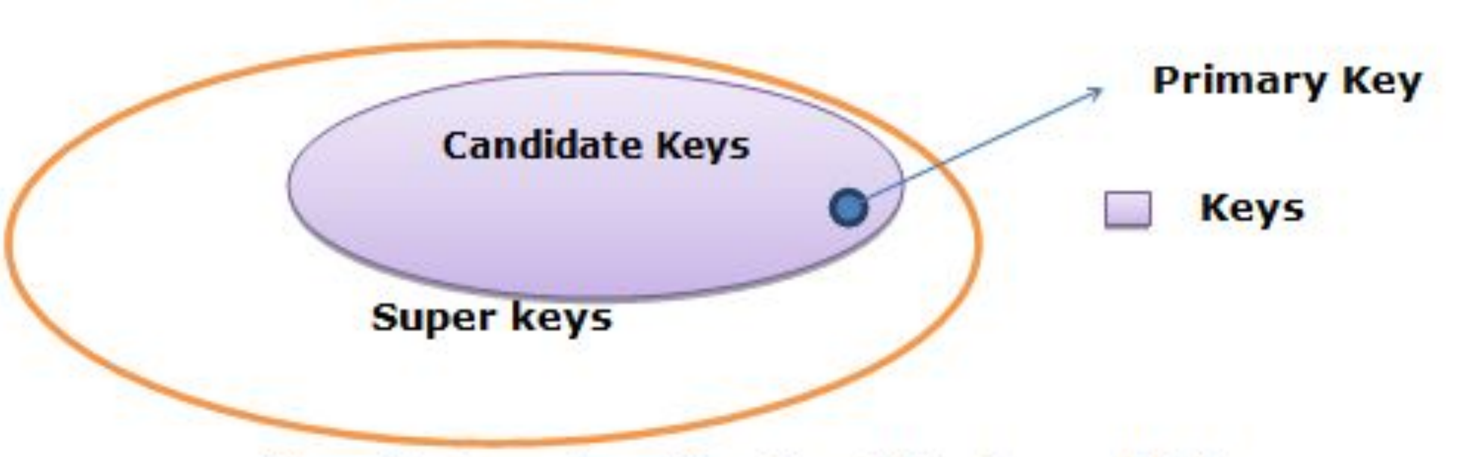




Key	• Data item that allows us to uniquely identify individual occurrences or an entity type.
Superkey	• Attribute or set of attributes that uniquely identify a tuple.
Candidate key	• Minimal super key with the property of irredusability and uniqueness
Primary key	• An entity type may have one or more possible candidate keys, the one which is selected as primary key.
Composite key	• candidate key that consisting of two or more attributes
Foreign key	• An attribute or set of attribute that matches the candidate key or other or same relation

Keys


Relational Keys
Super Key
Foreign Key
Candidate Key
Primary Key
Alternate Key


Primary Key
Candidate Keys
▨ Keys
Super keys

Relational Algebra & Relational Calculus

- Relational Algebra & Relational Calculus are 2 languages for Relational Model
- These languages help in data manipulation

Relational Algebra

- A procedural query language
- Basic set of operations for relational model
- Consists of a set of operations that take one or more relations as input & produce a new relation as output.

Relational Algebra & Relational Calculus

Relational Algebra (cont…)

- Operations in Relational Algebra

1. Fundamental operations

   a) **Select**: chooses a subset of tuples from relation that satisfies selection criteria. Its operator is sigma (σ). Example: $\sigma Dno = 4$ (employees) ⍰ select employees whose dno is 4

   $\sigma Salary > 40000$ (employees) ⍰ select employees whose salary is greater than 40000

   b) **Project**: selects some of the columns from table & discards other columns. It's used only if we are interested in some attributes. Result is vertical partition of relation into two columns (one with needed columns & next with discarded columns). Its operator is ($\prod$) . Example:

   $\prod$ fname, lname, age (employee) ⍰ select fname, lname, age from employee

Relational Algebra & Relational Calculus

Relational Algebra (cont…)

c) **Set Difference**: These operators need same domain in both operand relations. It allows to find tuples that are in one relation but not in another. It is denoted by (-). Example: ∏ customer_name (depositor) – ∏ customer_name (borrower) ⍰ select all customers of bank who have an account but no loan.

d) **Cartesian Product**: This allows to combine information from any two relations. Its operator is (x) . Example:

r = borrower x loan ⍰ borrower.cusname, borrower.loanNo, loan.loanNo, loan.branchName, loan.amount

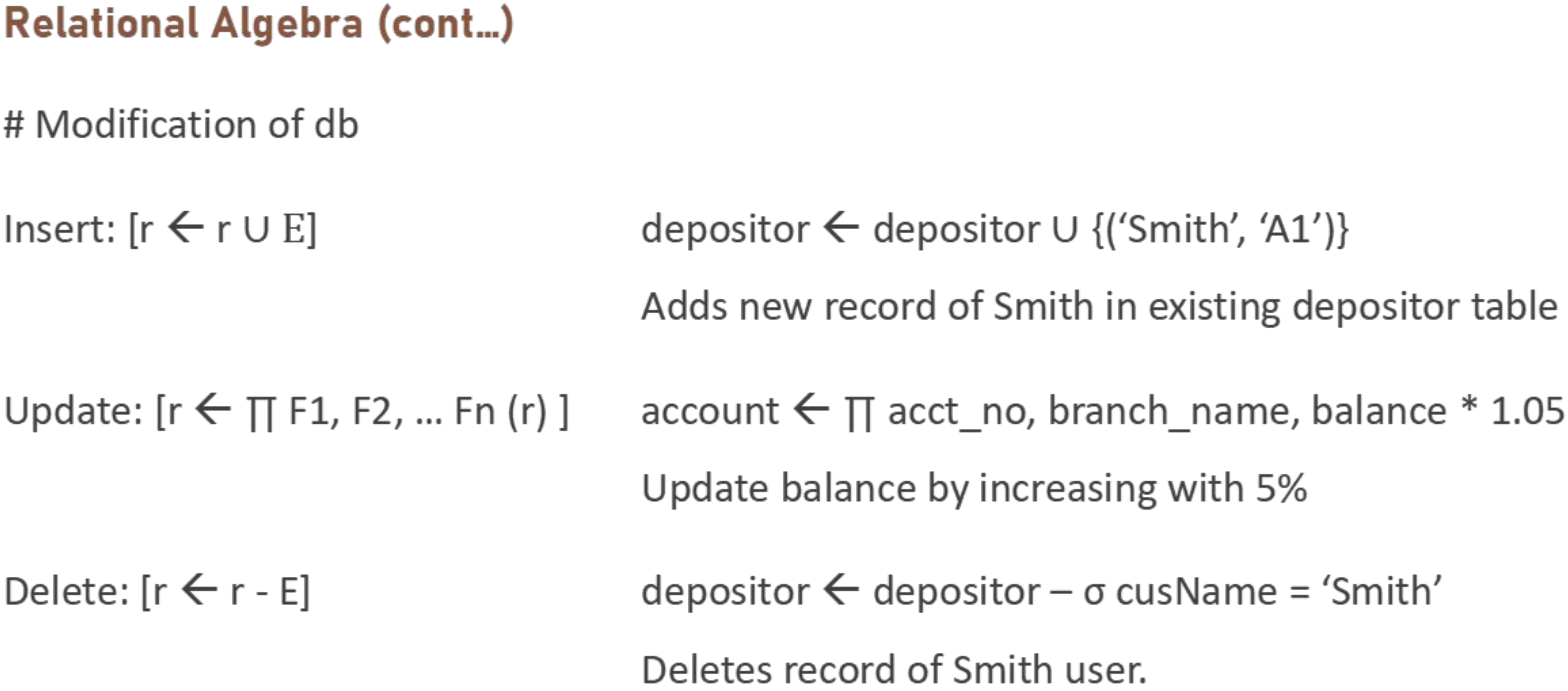
Relational Algebra & Relational Calculus

Relational Algebra (cont…)

2. Additional Operations:
   - i. Rename (ρ) = denoted by rho, renames objects
   - ii. Set intersection (∩) = getting common tuples appearing in both relations
   - iii. Natural Join (⋈) = mixing all columns from both relations
   - iv. Division (÷) = getting remaining tuples leaving from divided relation
   - v. Assignment (←) = applying new operation to that particular relation

3. Aggregate Functions:
   - Takes a collection of values & return a single value as a result
   - Example: sum, count, max, min, avg

Relational Algebra & Relational Calculus



Relational Algebra (cont…)

Modification of db

Insert: $[r \leftarrow r \cup E]$

$\text{depositor} \leftarrow \text{depositor} \cup \{(\text{'Smith'}, \text{'A1'})\}$

Adds new record of Smith in existing depositor table

Update: $[r \leftarrow \prod F1, F2, \ldots Fn\ (r)\ ]$

$\text{account} \leftarrow \prod \text{acct\_no}, \text{branch\_name}, \text{balance} * 1.05$

Update balance by increasing with 5%

Delete: $[r \leftarrow r - E]$

$\text{depositor} \leftarrow \text{depositor} - \sigma\ \text{cusName} = \text{'Smith'}$

Deletes record of Smith user.

Relational Algebra & Relational Calculus

Relational Calculus

- Alternative to relational algebra
- Is based on a branch of mathematical logic called Predicate Calculus
- Is non-procedural and is descriptive
- Describes the problem
- Allows to describe the set of answer **without being explicit about how they should be computed**
- Every calculus expression has equivalent algebraic expression
- Types:
- a) Tuple Relational Calculus b) Domain Relational Calculus

Relational Algebra & Relational Calculus

Relational Calculus (cont…)

1. Tuple Relational Calculus
   - Describes the info without giving a specific procedure for obtaining that info
   - Serves as theoretical basis for SQL
   - Variable takes tuple as value
   - Based on rows
   - Basic form is { T | P(T) } where T = a tuple variable, P(T) = formula that describes T
   - Ex: find branchName, loanNo, amount for loans of over 1200 price

     { t | t ϵ loan ^ t [amount] > 1200 }

Relational Algebra & Relational Calculus

Relational Calculus (cont…)

2. Domain Relational Calculus
   - Serves as theoretical basis of QBEL (Query By Example Language)
   - Uses domain variables that take on values from an attribute domain rather than values from an entire tuple
   - Based on columns
   - Basic form is { <x1, x2, …, xn> | p (<x1, x2, …, xn>) } where xi is either a domain variable or a constant, p(<x1, x2…>) is formula
   - Ex: find branchName, loanNo, amount for loans of over 1200

     { <l,b,a> | <l,b,a> ϵ loan ^ a > 1200}

Than

k

you!