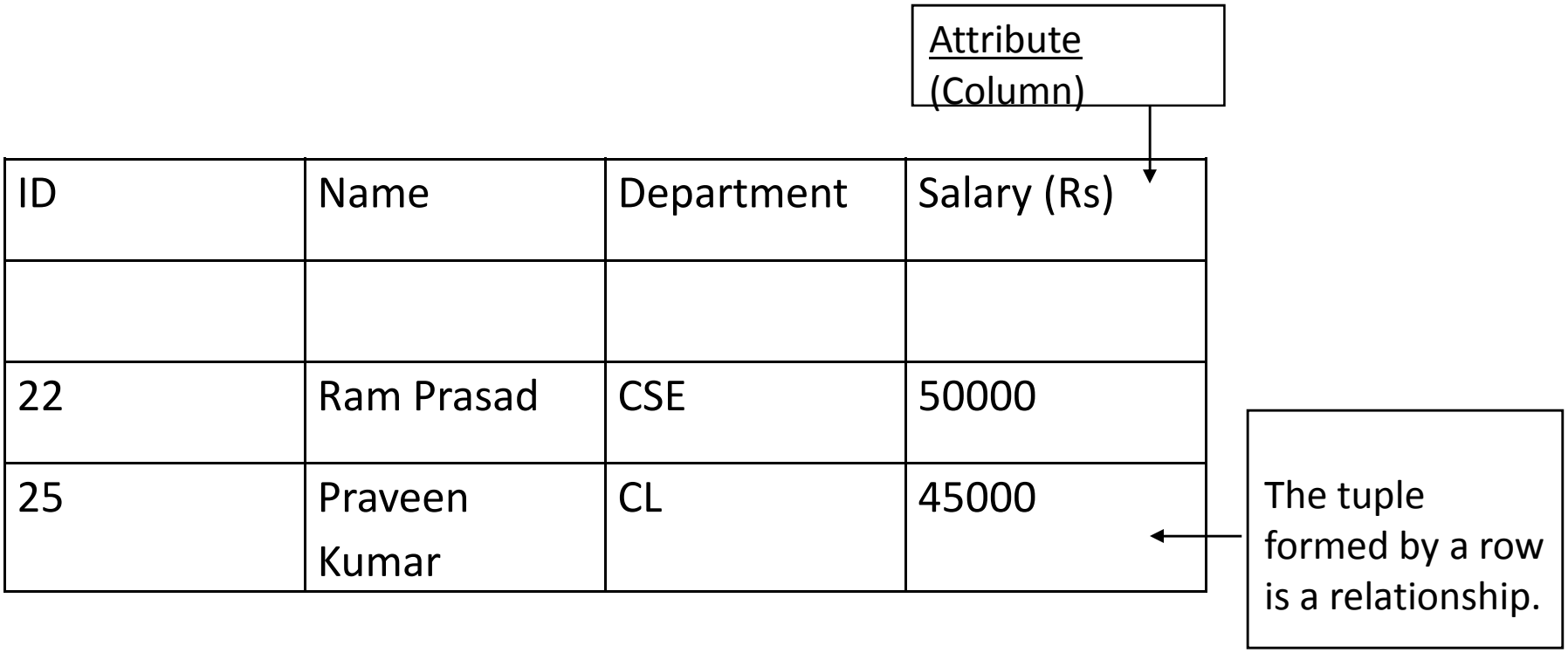
Lecture Notes for 3rd August 2011

Lecture topic : Introduction to Relational Model

Rishi Barua – 09010141
Shubham Tripathi – 09010148

Example of a **relation**.



Definitions

Domain of the Attribute: The set of permitted values for the attribute. Eg: For the attribute 'Department Name' the set of all department names is the domain.

Atomic Domain: A domain is atomic if elements of the domain are considered to be indivisible units. Eg: The set of names is non-atomic if we consider it to have first name ,middle name and last name , however age is an atomic unit.

Null value : An attribute takes a **null** value when an entity does not have a value for it. The *null* value may indicate "not applicable"—that is, that the value does not exist for the entity. For example, one may have no middle name. *Null* can also designate that an attribute value is unknown. The null value adds complications in the definition of several operations.

Formal Definition of a Relation

➔ Let A_1 ,A_2 , A_3, A_4 …..A_n be the attributes of the domain, then a= (A_1 ,A_2 , A_3, A_4 …..A_n)
is called a relation schema. A relation 'R' is a subset of the cartesian product D_1 X D_2 X D_3 ….X D_N , where D_1 , D_2 , D_3 ….D_N are the domain set of the attributes A_1 ,A_2 , A_3, A_4 …..A_n .
A relation instance(current values) of a relation is specified by a table.
Relations are **unordered** ,i.e order of tuples does not matter an can be stored in any arbitrary order.

➔Repetition of information and need for null values result in bad design .
➔Normalization theory is used for generating good designs.

➔A database consists of multiple relations.

Keys

A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely an entity in the entity set. If *K* is a superkey, then so is any superset of *K*. We are often interested in superkeys for which no proper subset is a superkey. Such minimal superkeys are called **candidate keys**.

The term **primary key** denotes a candidate key that is chosen by the database designer as the principal means of identifying entities within an entity set. The primary key should be chosen such that its attributes are never, or very rarely, changed. For instance, the address field of a person should not be part of the primary key, since it is likely to change. The student roll no. or id is an example of a primary key as it is highly unlikely to change and uniquely identifies the student.

Foreign key- A foreign key is an attribute of a different entity that is used to identify the current entity. The changes affected on the foreign key in the initial entity set ,will automatically be updated in the current entity set. For eg: if the student's department changes it will be updated in every set , where the student's department is a foreign key.

It results in a **referencing relation** and a **referenced relation**.

QUERY LANGUAGE

Query Languages is a language in which a user requests information from the database. They can be categorised as :

➔Procedural Language

➔Non procedural or Declarative Language

1. In a procedural language the user instructs the system to perform a system of operations on the database to compute the desired results. Eg: Relational Algebra
2. In a non-procedural language the user describes the information desired without giving a specific procedure for obtaining that information. Eg: Tuple Related Calculus
3. Most commercial relational database system offer a query language that includes elements of both the procedural and the non-procedural approaches.

Examples of pure languages:

Relational Algebra, Tuple Related Calculus, Domain Related Calculus.

Fundamental Operations

The Select Operation

The select operation selects tuples that satisfy a given predicate. We use the 'σ' to denote selection. The predicate appears as a subscript to σ.
Eg: σ_{name}="tree" , $\sigma_{loan} > 1200$
The select operation can also be used to select attibutes(columns) , as in Select Name and Roll No. from student table which contains various attributes of the student.

Composition of Relational Operations

Composing relational algebra operations into relational algebra expessions is just like composing arithmetic operations (such as +,-,/,*)

Union Operation

This involves the union of two search tuples .Eg: Selecting plants who are either violets or roses. i.e. it combines the two sets of roses and violets.

The Cartesian Product Operator

The cartesian product operation denoted by X allows us to combine information from any two relations. We write the cartesian product of relations R_1 and R_2 as R_1XR_2

Set difference operation

It is denoted by - , and allows us to find tuples that are in one relation but are not in another. The expression r-s results in a relation containing those tuples in r but not in s.

Additional Operations

Set intersection operation

It is denoted by Π. It is not a fundamental operation and does not ad any power to the relational algebra. r Π s = r – (r –s)

Natural Join Operation

The natural join is a binary operation that allows us to combine certain selections and the cartesian product into one operation. The natural join operation forms a cartesian prduct of its two arguments, performs a selection forcing equality on those attributes that appear in both relation schemas and finally removes duplicate attributes.
Let *r* and *s* be relations on schemas *R* and *S* respectively.
Then, the "natural join" of relations *R* and *S* is a relation on schema $R \cup S$ obtained as follows:

Consider each pair of tuples t_r from *r* and t_s from *s*.
If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple *t* to the result, where

t has the same value as t_r on *r*

t has the same value as t_s on *s*

➔ $r \bowtie s$

Reference : Codd's 12 rules : http://en.wikipedia.org/wiki/Codd%27s_12_rules