Course code : **CSE2007**
Course title : **Database Management System**
Module : **3**
Topic : **1**

Relational Operations

Objectives

This session will give the knowledge about

- Relational Operations
- Relational Algebra

Relational Query Languages

A query language is a language in which a user requests information from the database. It can be categorized as either procedural or nonprocedural.

Procedural language:

> The user instructs the system to perform a sequence of operations on the database to compute the desired result. Example: Relational Algebra

Non procedural language:

> The user describes the desired information without giving a specific procedure for obtaining that information. Example: Tuple relational calculus and Domain relational calculus

Relational Algebra

An algebra, in general, consists of operators and atomic operands.

For instance, in the algebra of arithmetic:

- The atomic operands are variables like x and constants like 15.
- The operators are : addition, subtraction, multiplication, and division.
- For instance: arithmetic expressions (x + y) * z or ((x + 7)/(2/ - 3)) + x.

Relational algebra is another example of an algebra. Its atomic operands are:

1. Variables that stand for relations.
2. Constants, which are finite relations.

Relational Algebra Operations

The operations of the Relational Algebra fall into four broad classes:

a) The usual set operations: union, intersection, and difference — applied to relations.

b) Operations that remove parts of a relation: “selection” eliminates some rows (tuples), and “projection” eliminates some columns.

c) Operations that combine the tuples of two relations, including “Cartesian product” which pairs the tuples of two relations in all possible ways, and various kinds of “join” operations, which selectively pair tuples from two relations.

d) An operation called “renaming” that does not affect the tuples of a relation, but changes the relation schema, i.e., the names of the attributes and/or the name of the relation itself.

Set Operations

The three most common operations on sets are union, intersection, and difference. We assume the reader is familiar with these operations, which are defined as follows on arbitrary sets R and S:

- R U S, the union of R and S, is the set of elements that are in R or S or both. An element appears only once in the union even if it is present in both R and S.
- R ∩ S, the intersection of R and S, is the set of elements that are in both R and S.
- R - S, the difference of R and S, is the set of elements that are in R but not in S. Note that R - S is different from S - R; the latter is the set of elements that are in S but not in R.

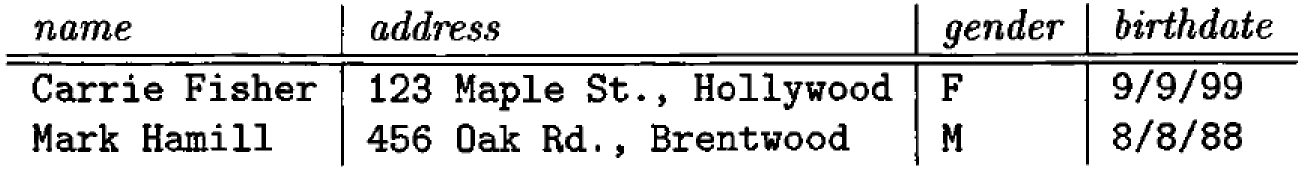
Conditions for Set Operations

When we apply these operations to relations, we need to put some conditions on R and S:

1. R and S must have schemas with identical sets of attributes, and the types (domains) for each attribute must be the same in R and S.

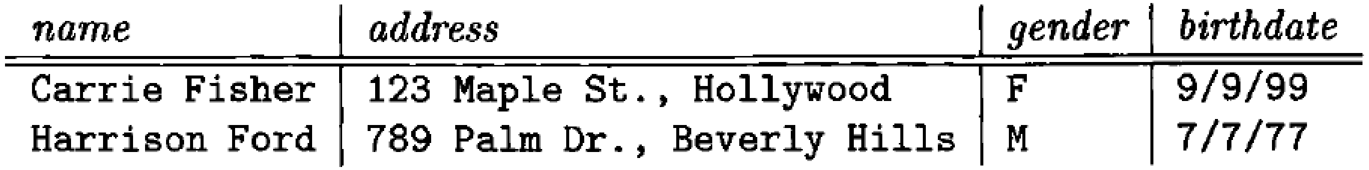
2. Before we compute the set-theoretic union, intersection, or difference of sets of tuples, the columns of R and S must be ordered so that the order of attributes is the same for both relations.

Set Operations - Examples



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

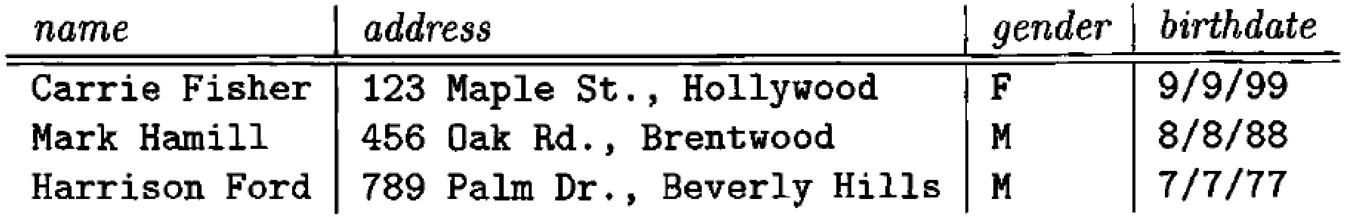
Relation R



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

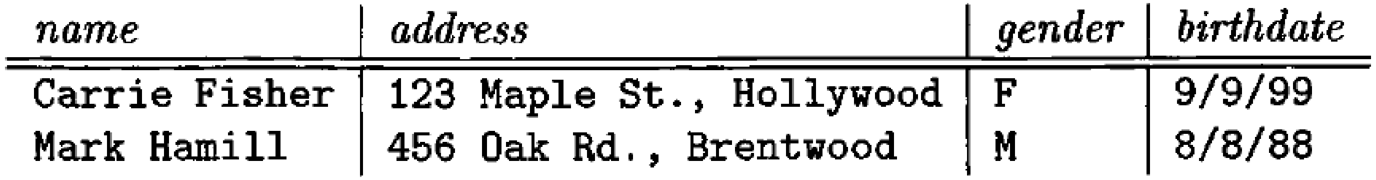
Relation S

R U S is



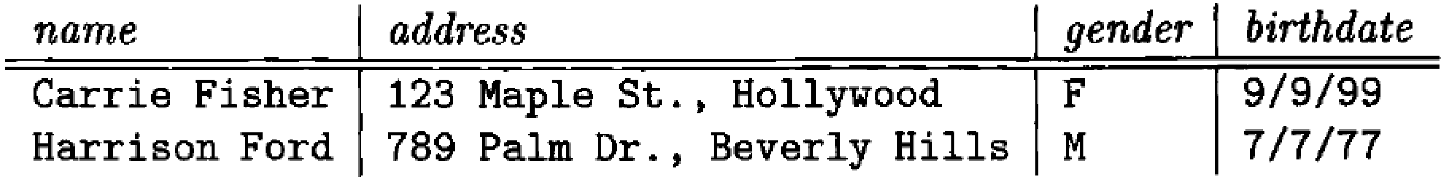
name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

Set Operations - Examples



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

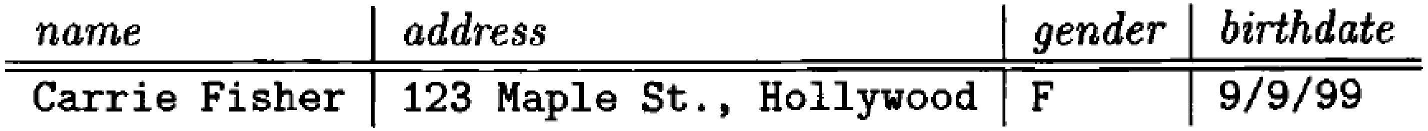
Relation R



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

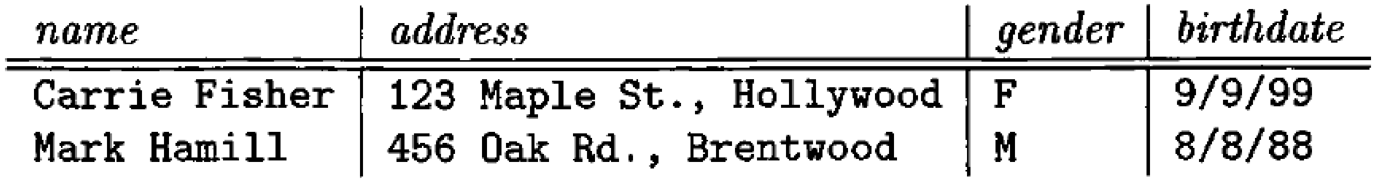
Relation S

R ∩ S is



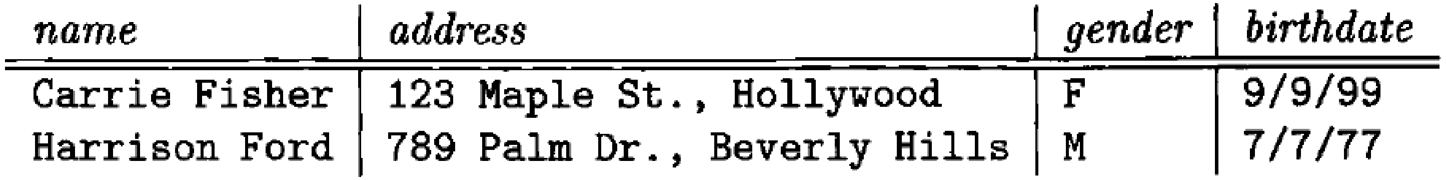
name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99

Set Operations - Examples



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

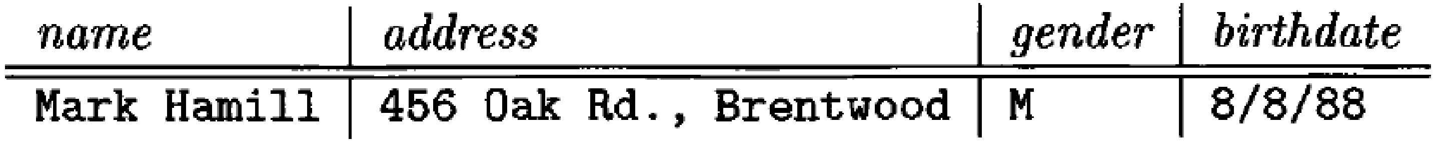
Relation R



name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

Relation S

R - S is

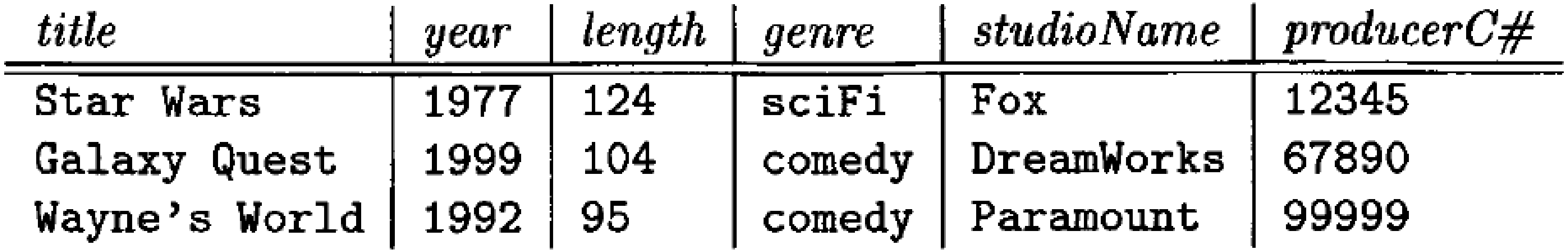


name	*address*	*gender*	*birthdate*
Mark Hamill	456 Oak Rd., Brentwood	M	8/8/88

Projection

The projection operator is used to produce from a relation R a new relation that has only some of R’ columns.

The relation Movies

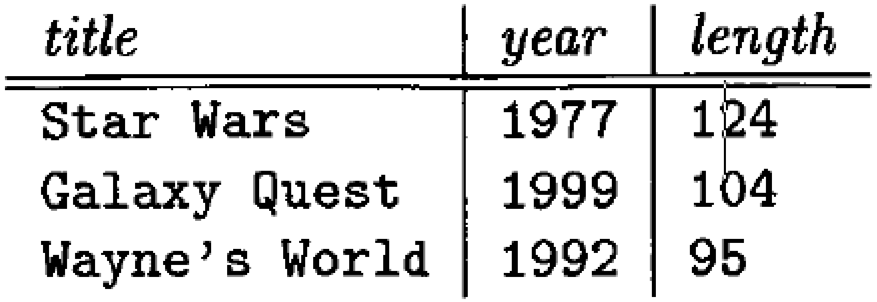


title	*year*	*length*	*genre*	*studioName*	*producerC#*
Star Wars	1977	124	sciFi	Fox	12345
Galaxy Quest	1999	104	comedy	DreamWorks	67890
Wayne’s World	1992	95	comedy	Paramount	99999

Projection

$\pi_{(title,year,length)}(Movies)$

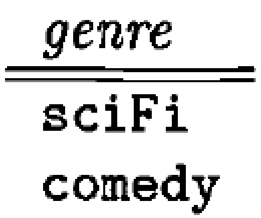
The resulting relation is



title	*year*	*length*
Star Wars	1977	124
Galaxy Quest	1999	104
Wayne's World	1992	95

$\pi_{(genre)}(Movies)$

The resulting relation is

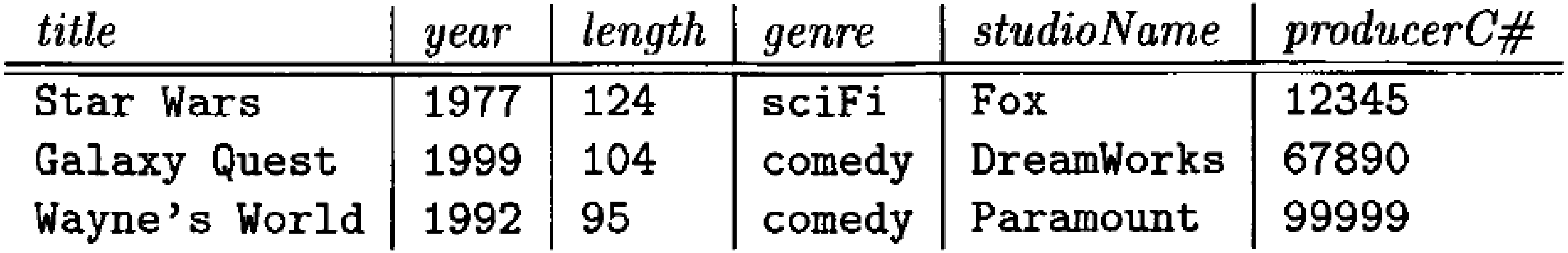


genre
sciFi
comedy

Selection

The selection operator, applied to a relation R, produces a new relation with a subset of R’s tuples. The tuples in the resulting relation are those that satisfy some condition C that involves the attributes of R.

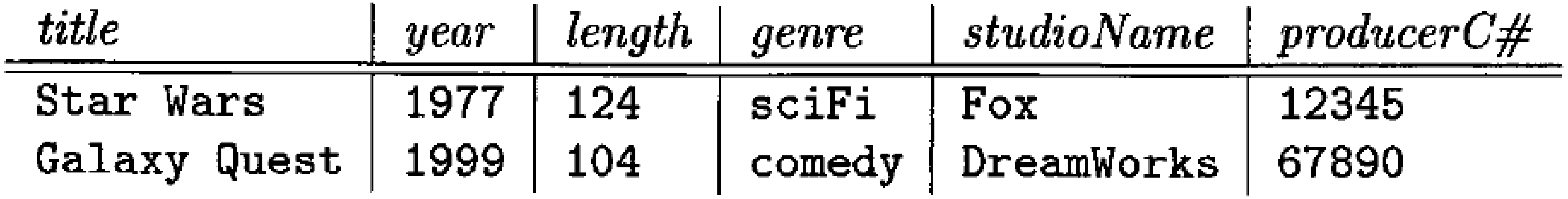
The relation Movies



title	*year*	*length*	*genre*	*studioName*	*producerC#*
Star Wars	1977	124	sciFi	Fox	12345
Galaxy Quest	1999	104	comedy	DreamWorks	67890
Wayne’s World	1992	95	comedy	Paramount	99999

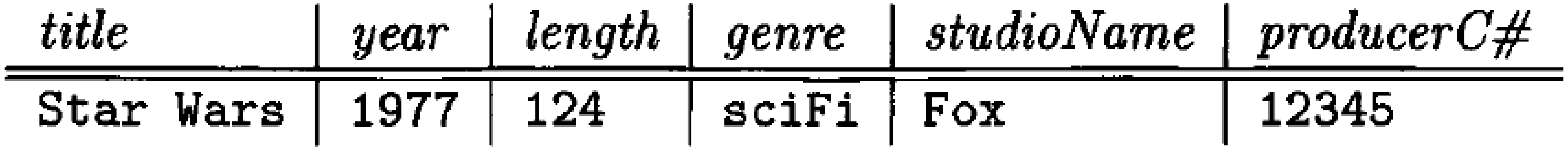
Selection

$\boldsymbol{\sigma_{(length \geq 100)}(Movies)}$ - The resulting relation is



title	year	length	genre	studioName	producerC#
Star Wars	1977	124	sciFi	Fox	12345
Galaxy Quest	1999	104	comedy	DreamWorks	67890

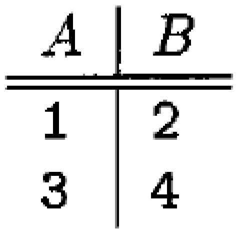
$\boldsymbol{\sigma_{(length \geq 100\ AND\ studioName = "Fox")}(Movies)}$ - The resulting relation is



title	year	length	genre	studioName	producerC#
Star Wars	1977	124	sciFi	Fox	12345

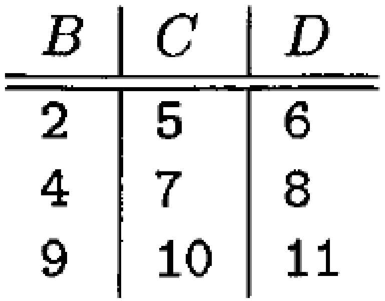
Cartesian Product

The Cartesian product (or cross-product, or just product) of two sets R and S is the set of pairs that can be formed by choosing the first element of the pair to be any element of R and the second any element of S. This product is denoted R x S.



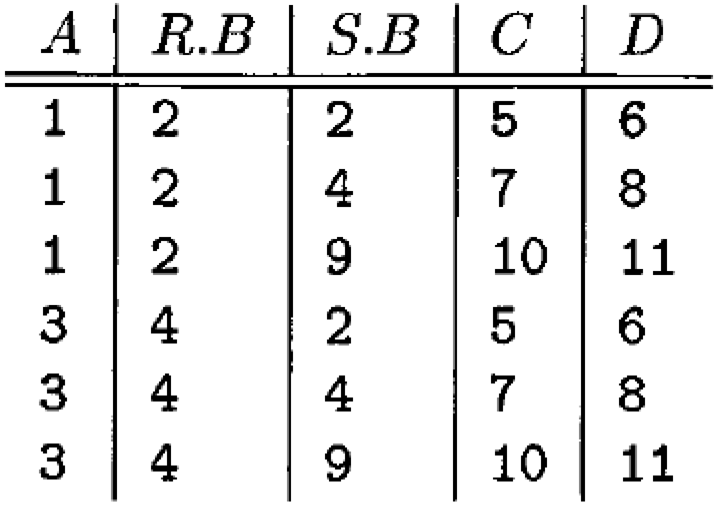
A	B
1	2
3	4

(a) Relation R



B	C	D
2	5	6
4	7	8
9	10	11

(b) Relation S



A	$R.B$	$S.B$	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

(c) Result $R \times S$

Summary

This session will give the knowledge about

- Relational Operations
- Relational Algebra