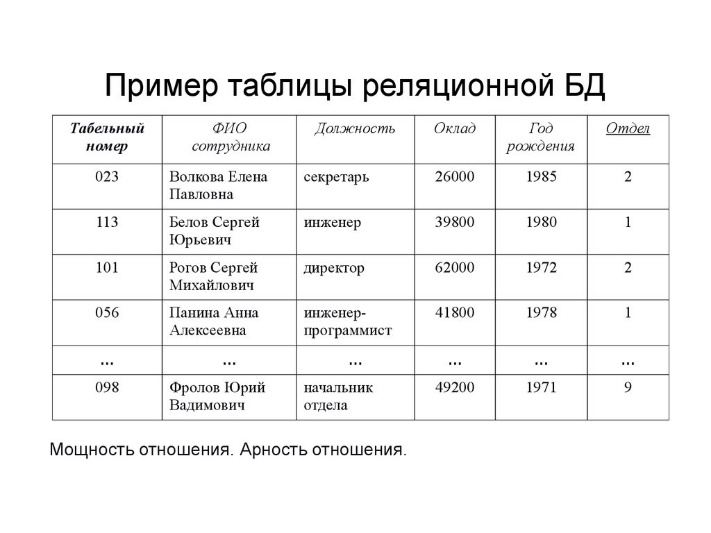
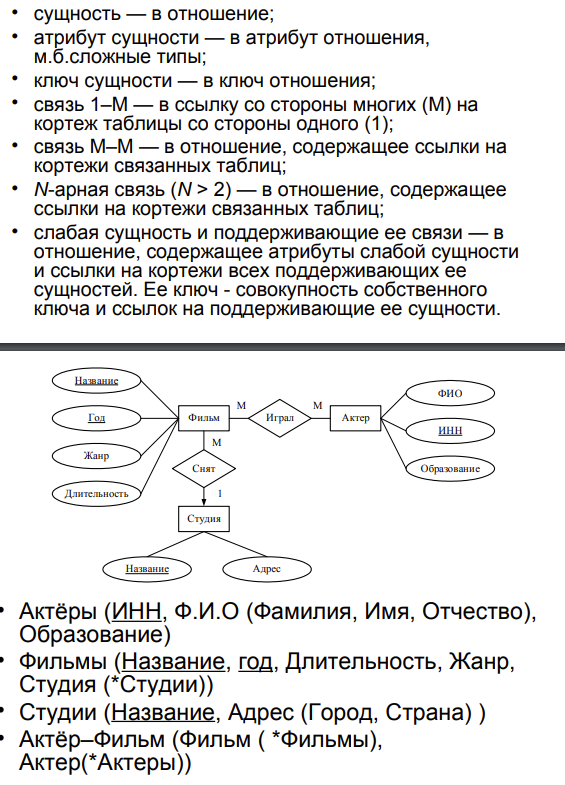
**Схема в реалиционной модели**

****

**Схема в UDT (объектно-реляционной модели)**

****

Связь ISA задают через наследование типов

****

**В объектной модели (ODL)**

Class Фильм ( extent Фильмы key (Год, Название) )

{

attribute string Название;

attribute integer Год;

attribute Bag<string> Награды;

attribute Struct{ string Режиссер,

List<string> Ассистенты} Создатели;

relationship Set<Actor> acts

inverse Actor::infs;relationship Studia stud

inverse Studia::fs;

}

Пример описания связи 1-М

class Фильм (...)

{

.....

relationship Студия made

inverse Студия::owns;

}

class Студия (...)

{

...

relationship Set<Фильм> owns

inverse Фильм::made

}

Сигнатура содержит название метода, тип возвращаемого

значения, перечень параметров и перечень исключений,

например:

string GetActorName(in integer RoleTime)

raises (noAct,manyAct)

• где raises — перечисляет исключения, вызываемые методом.

• При описании параметров метода указывают способ их

передачи:

– In — по значению (входной), по умолчанию;

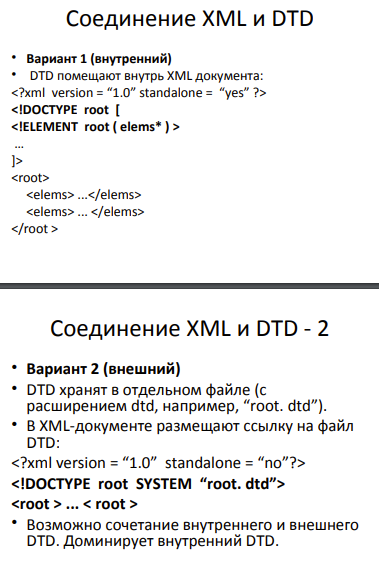
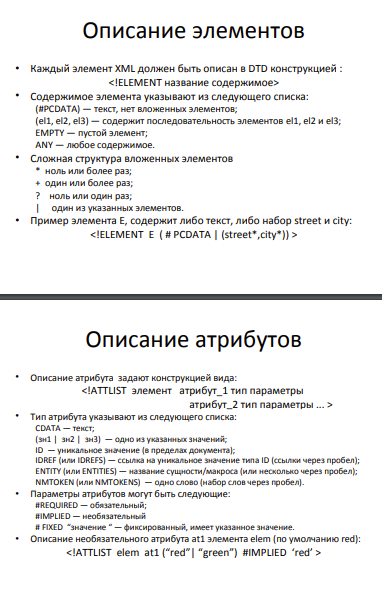
– Inout — по ссылке;

– Out — по ссылке (выходной).

• Тело метода описывают при реализации в среде

программирования.

**XML и DTD**



Пример DTD

<!ELEMENT db (film\*, actor\*, studio+)>

<!ELEMENT film (name, year, len?, type?)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT year (#PCDATA)>

<!ELEMENT len (#PCDATA)>

<!ELEMENT type (#PCDATA)>

<!ELEMENT actor (fio, edu+)>

<!ELEMENT fio (#PCDATA)>

<!ELEMENT edu (#PCDATA)>

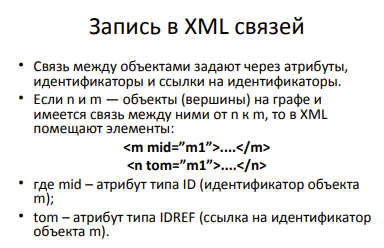
<!ELEMENT studio (name, addr )>

<!ELEMENT addr (#PCDATA)>

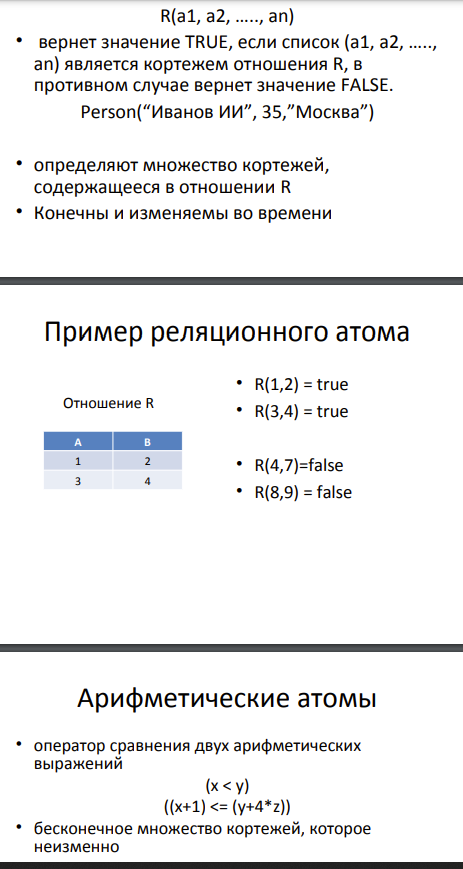
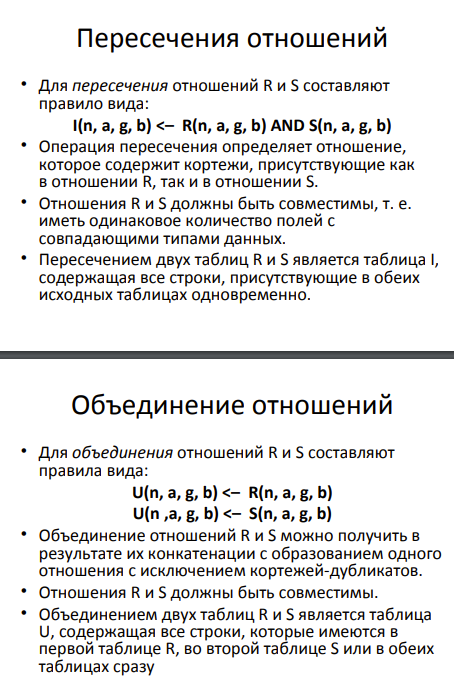
<!ATTLIST film idf ID #REQUIRED toact IDREFS tos IDREF>

<!ATTLIST actor ida ID tof IDREFS>

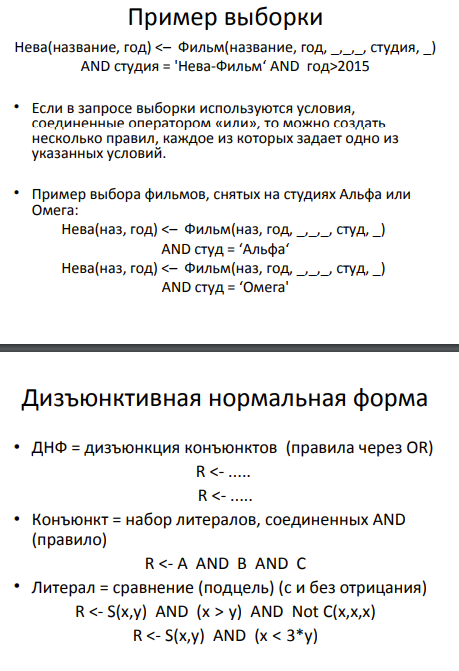
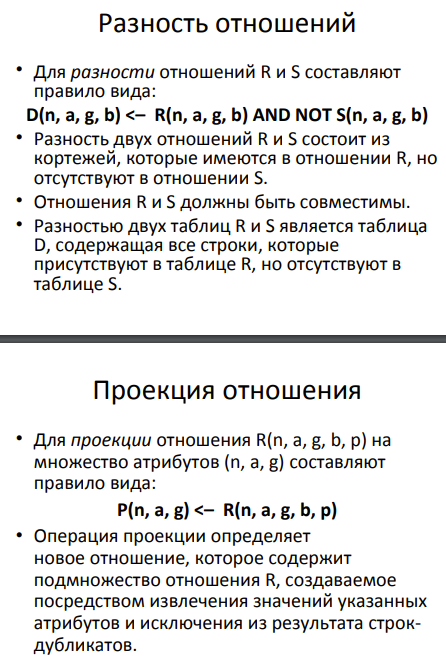
<!ATTLIST studio ids ID tof IDREFS>

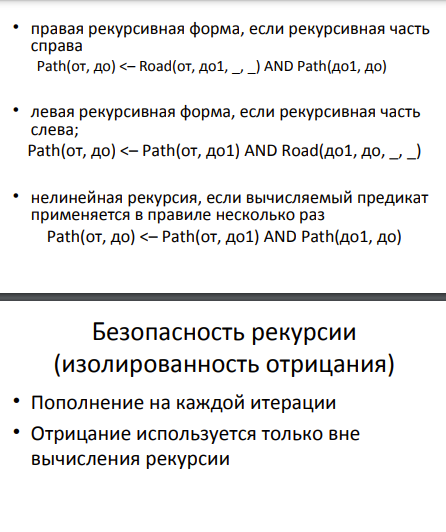
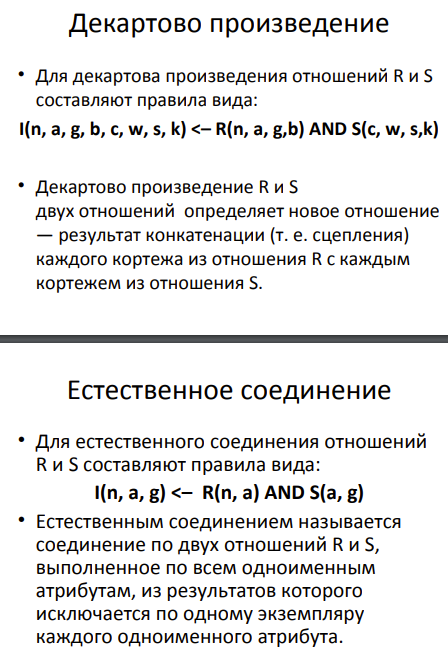


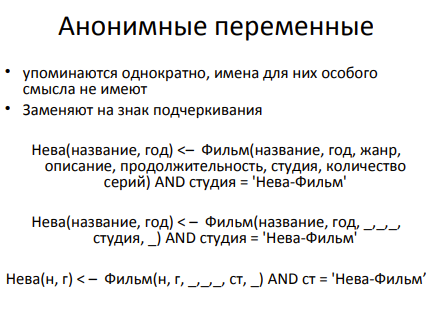
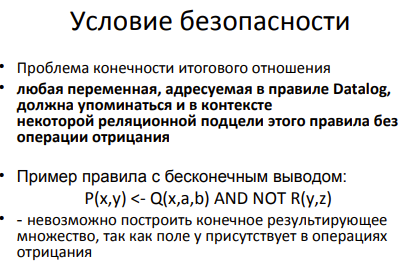
**Datalog для РБД**

<> (не равно)





Пример запроса 1

• Фильмы, снятые на студии Мосфильм:

ФильмМосфильм(н,г) <– Фильм(н, г, \_, \_, ст)

AND ст = "Мосфильм"

Пример запроса 2

• Актёры, работавшие в 1990г. на студии

Мосфильм:

АктМосфильм(ФИО) <– Актёр(i, ФИО, \_)

AND ФА(н, 1990, i)

AND Фильм(н, 1990, \_, \_, "Мосфильм")

Пример запроса 3

• Актёры, никогда не работавшие на студии

Мосфильм:

АктНикогдаМосфильм(ФИО) <–

Актёр(i, ФИО, \_)

AND NOT ФА(н, г, i)

AND Фильм(н, г, \_, \_, "Мосфильм")

Пример запроса 4

• Студия, снявшая более одного фильма:

СтБольшеОдногоФильма(ст) <–

Фильм(н, г, \_, \_, ст)

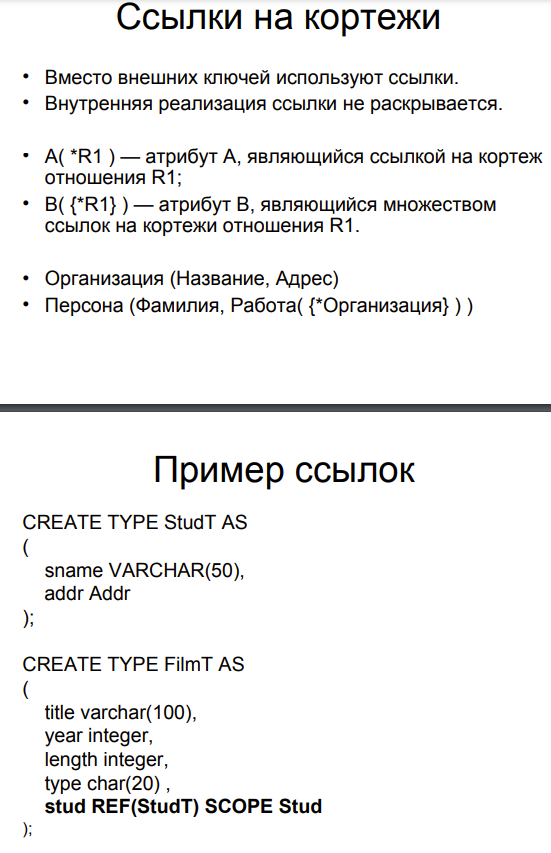
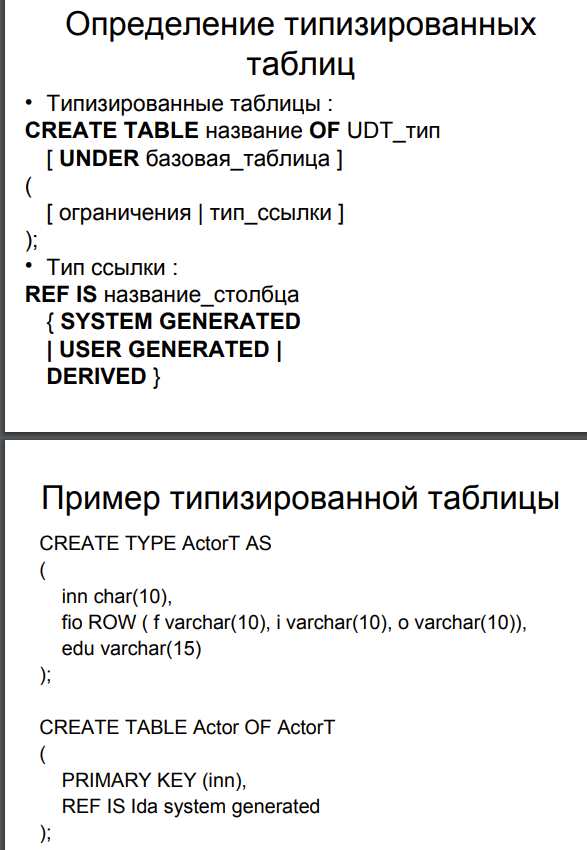
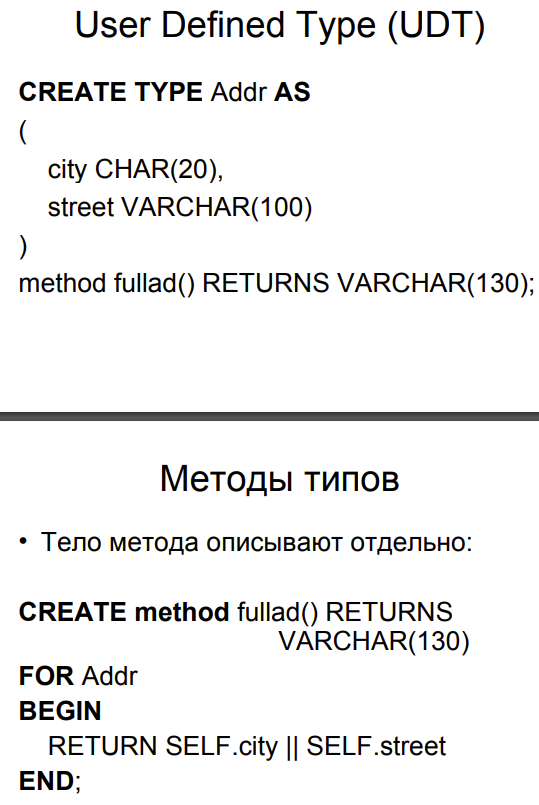
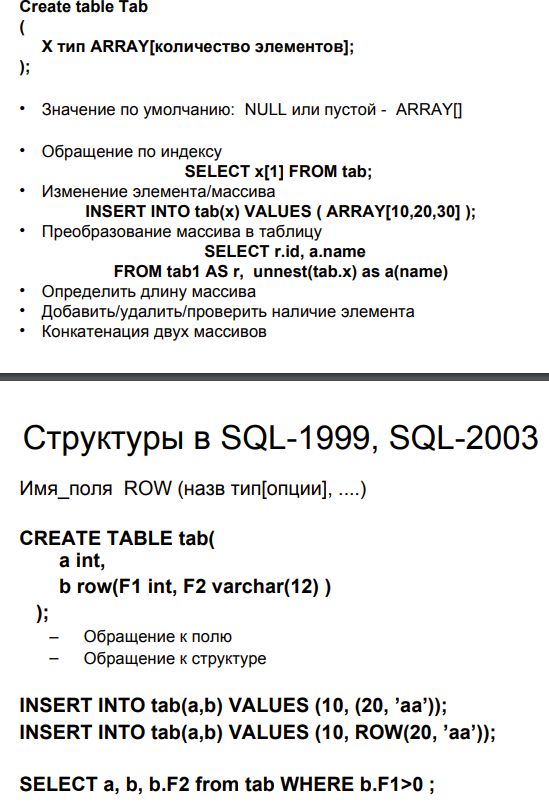
AND Фильм(н2, г2, \_, \_, ст)

AND (н2 != н)

**SQL-1999 для UDT**

Обычный sql

FOREIGN KEY (имя\_столбца\_внешнего\_ключа [,...n ] ) REFERENCES имя\_родительской\_таблицы

****

Разыменование ссылки

SELECT deref( act )

FROM FA

WHERE film–>title='The Matrix'

Использование методаобозревателя

• Нельзя обращаться к атрибутам по конкретным именам,

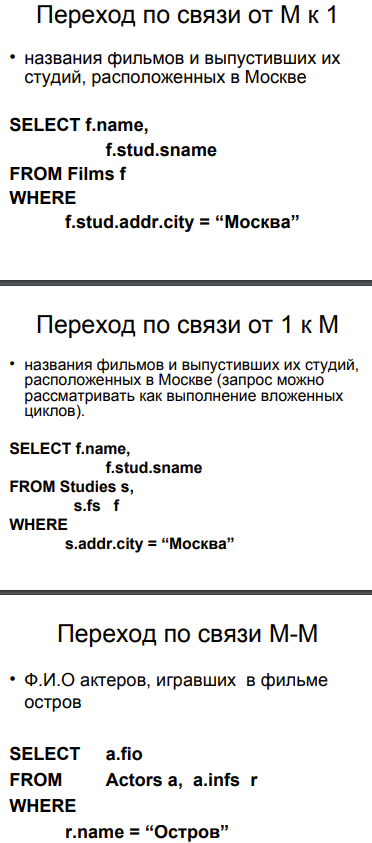
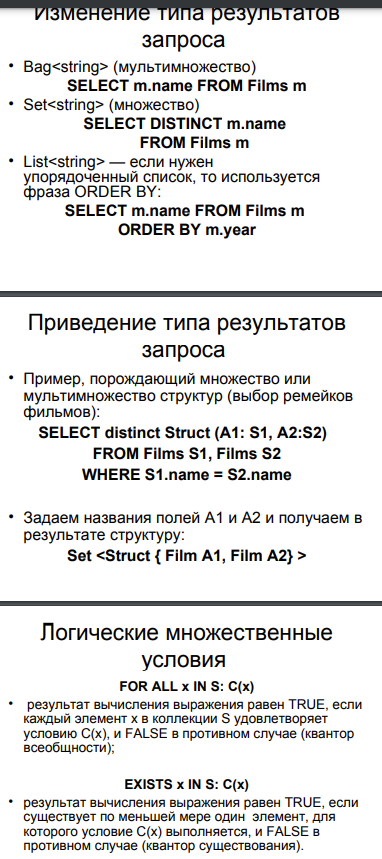
• вместо этого используют метод-обозреватель (он вызывается при обращении к атрибуту)

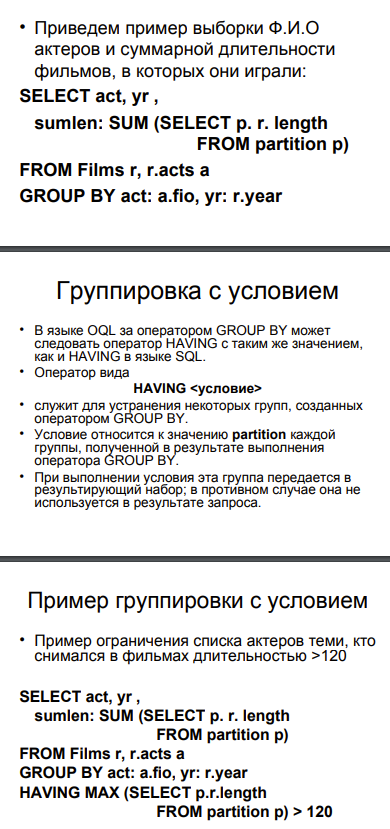
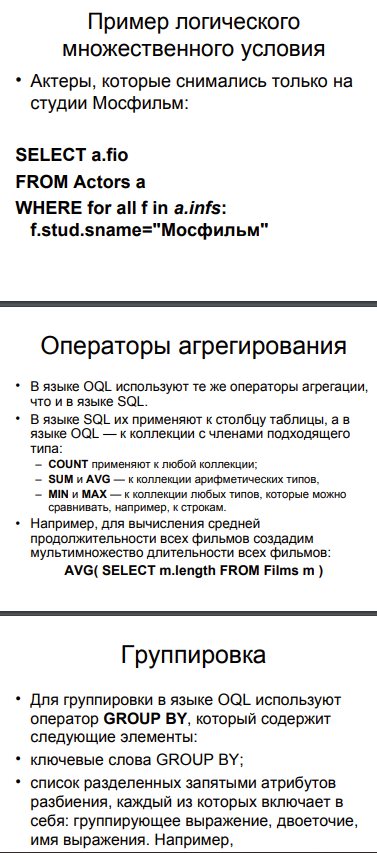
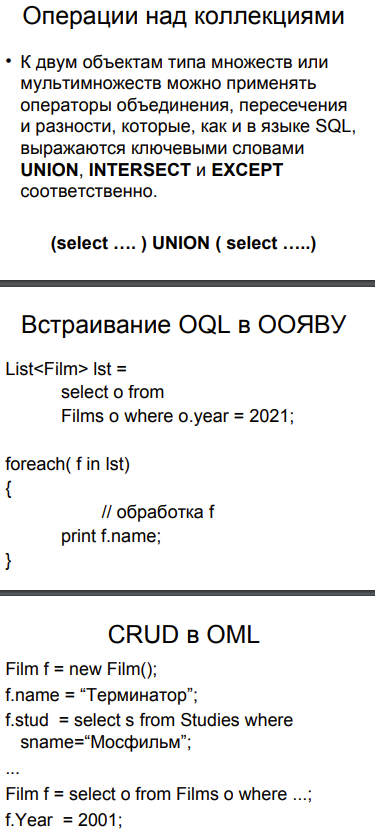
SELECT title()

FROM TF

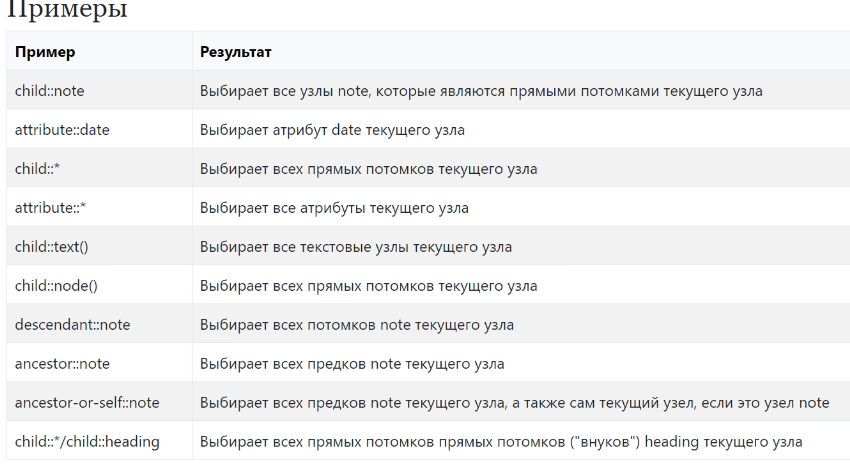
WHERE type() < > ‘комедия

**OQL для классов**

**Xpath**



1) \* — Выбрать любой элемент.

2) [] — Найти конкретный элемент. Пример: //li[1]имя\_узла — Выбирает все узлы с указанным именем узла. Пример: div, p и т.д.

3) / — Ищет от корневого узла html

4) // — Ищет узлы в документе от текущего узла, который соответствует выбору, независимо от того, где они находятся

5) . — Ищет текущий узел

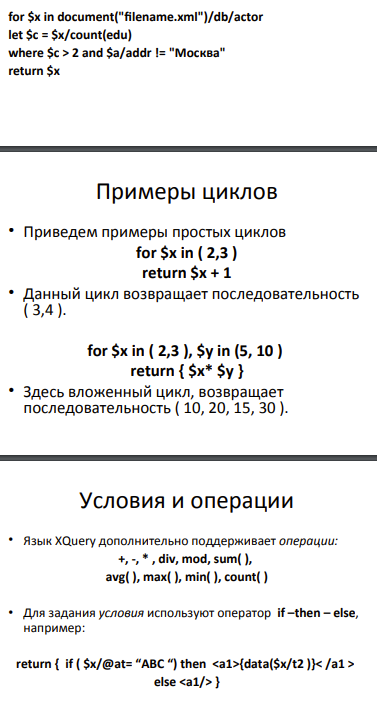
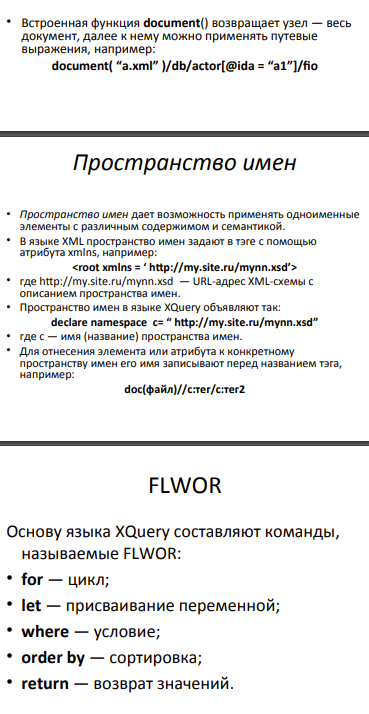
6) .. — Ищет родителя текущего узла

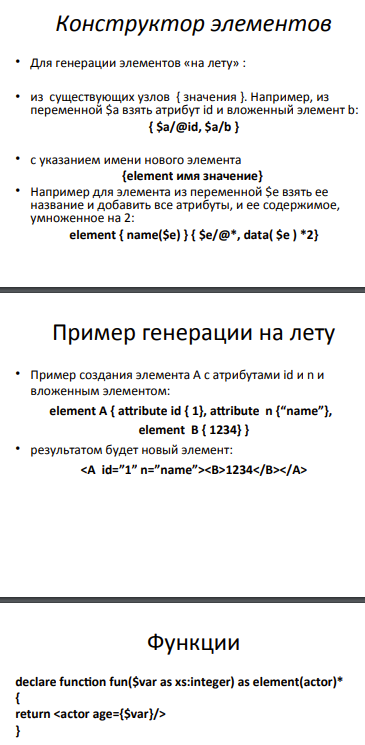
7) @ — Ищет нужный атрибут. Пример: //p[@value="2008"]

//actor|//studio – объединение нескольких путей

//fio [.=”Иванов”] – элемент fio значение, которого равно Иванов

**Xquery**





Запрос, который ищет актеров, снявшихся более чем в

трех фильмах, и возвращает Ф.И.О актера, его 1-ое

образование и количество его фильмов с сортировкой по

Ф.И.О:

for $a in document( “filename.xml”)//actor

let $f:= //film[ contains($a/@tof, @fid) ]

where count($f) >3

order by $a/fio

return { $a/fio, $a/edu[1], { count ($f ) } }

Запрос на формирование списка Ф.И.О всех актеров:

< ul >

{

for $x in doc ( “filename.xml”)//fio/text ( )

order by $x

return <li>{ $x} </li>

}

< ul >