



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

Базы данных

Лекция 2 Реляционные БД

Агафонов Антон Александрович
д.т.н., доцент кафедры ГИИБ

Самара



- Основы реляционных БД:
 - тип данных
 - домен
 - атрибут
 - кортеж
 - отношение
- Фундаментальные свойства отношений
- Целостность данных
- Основные понятия реляционной алгебры





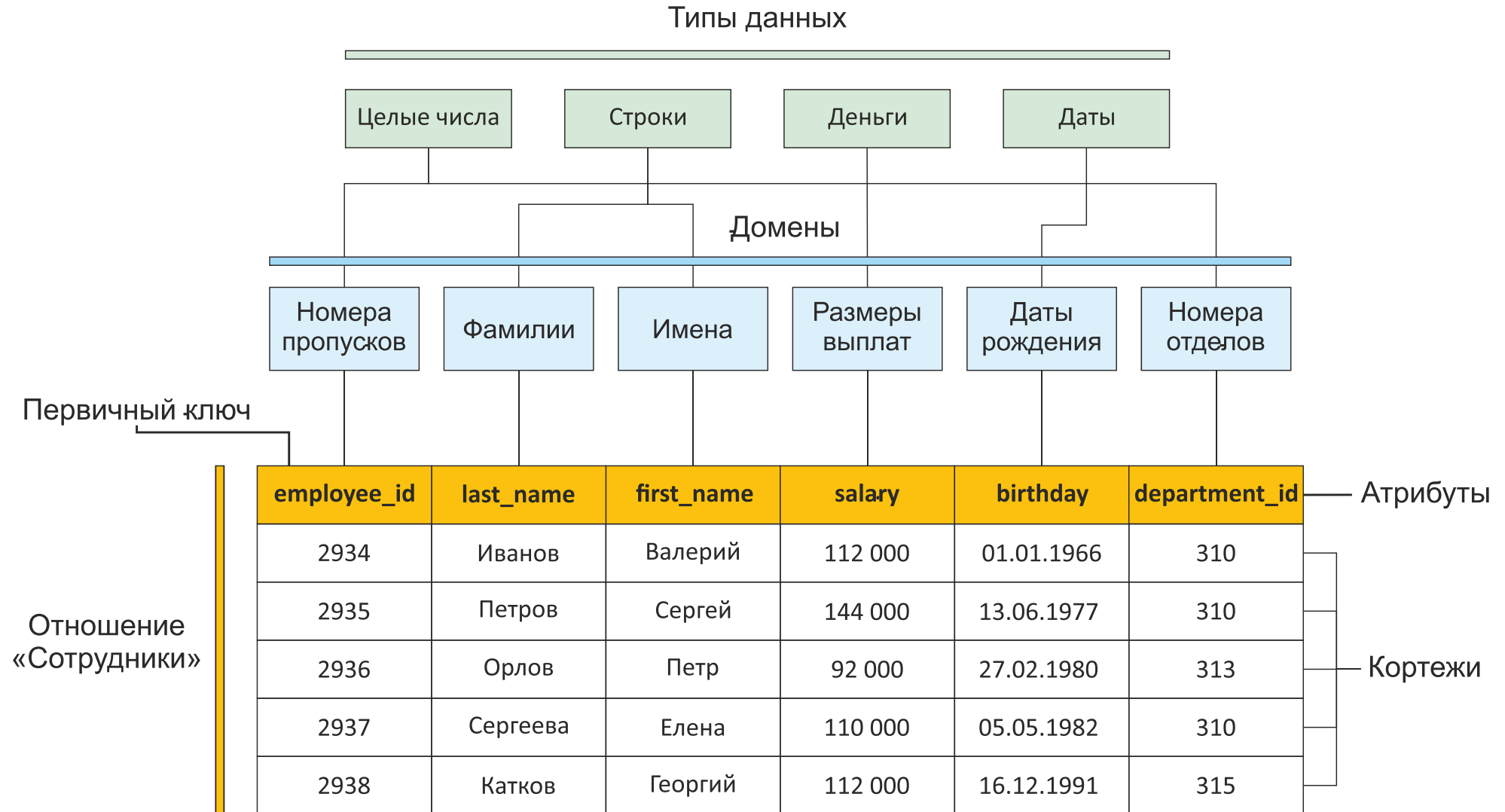
Достоинства:

- ▲ Наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей и допускают точные формальные определения, оставаясь интуитивно понятными
- ▲ Наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации баз данных
- ▲ Возможность манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти





Базовые понятия реляционных БД





Основные типы данных:

- Числовые: целые, дробные с плавающей точкой, дробные с фиксированной точностью
- Темпоральные: дата, время, временной интервал
- Символьные
- Байтовые
- Пространственные
- Слабоструктурированные: JSON, XML
- Расширенные

Тип данных интегрирует в себе компоненты:

- ограничение множества принадлежащих типу значений
- определение применяемых к типу набора операций
- определение способа отображения (внешнего представления) значений типа





- Домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.
- Домен – допустимое потенциального множества значений данного типа.
- Данные считаются сравнимыми только в том случае, когда они относятся к одному домену.





Отношение R состоит из *схемы* и *тела*.

Схема H отношения R — конечное множество упорядоченных пар вида (A_i, T_i) , где A_i — имя атрибута, а T_i — имя домена, $i = 1, \dots, n$. По определению требуется, чтобы все имена атрибутов в заголовке отношения были различными (уникальными).

Тело B отношения R — множество кортежей t .

Кортеж t , соответствующий схеме H — множество упорядоченных пар вида (A_i, v_i) , где v_i — допустимое значение домена T_i .

Количество кортежей называют кардинальным числом отношения (кардинальностью), или **мощностью** отношения.

Количество атрибутов называют степенью, или «**арностью**» отношения





Атрибут
(столбец, поле)

Имя столбца

Отношение
(таблица)

employee_id	last_name	first_name	salary	birthday	department_id
2934	Иванов	Валерий	112 000	01.01.1966	310
2935	Петров	Сергей	144 000	13.06.1977	310
2936	Орлов	Петр	92 000	27.02.1980	313
2937	Сергеева	Елена	110 000	05.05.1982	310
2938	Катков	Георгий	112 000	16.12.1991	315

Кортеж
(строка, запись)





1. Отсутствие кортежей-дубликатов

Дублирование данных противоречит теории множеств, т.к. в классической теории множеств по определению каждое множество состоит из различных элементов.

2. Отсутствие упорядоченности кортежей

Свойство отсутствия упорядоченности кортежей отношения является следствием определения отношения-экземпляра как множества кортежей. Отсутствие требования к поддержанию порядка на множестве кортежей отношения дает дополнительную гибкость СУБД.





3. Отсутствие упорядоченности атрибутов

Атрибуты отношений не упорядочены, поскольку по определению схема отношения есть множество пар $\langle A, T \rangle$. Для ссылки на значение атрибута в кортеже отношения всегда используется имя атрибута.

4. Атомарность значений атрибутов

Значения всех атрибутов являются атомарными. Это следует из определения домена как некоторого подмножества значений простого типа данных, т.е. среди значений домена не могут содержаться множества значений (отношения).

Принято говорить, что в реляционных базах данных допускаются только нормализованные отношения или отношения, представленные в первой нормальной форме.



Ненормализованное отношение

department_id	department				
	employee_id	last_name	first_name	salary	birthday
310	2934	Иванов	Валерий	112 000	01.01.1966
	2935	Петров	Сергей	144 000	13.06.1977
	2937	Сергеева	Елена	110 000	05.05.1982
313	2936	Орлов	Петр	92 000	27.02.1980
315	2938	Катков	Георгий	112 000	16.12.1991



Нормализованное отношение

employee_id	last_name	first_name	salary	birthday	department_id
2934	Иванов	Валерий	112 000	01.01.1966	310
2935	Петров	Сергей	144 000	13.06.1977	310
2936	Орлов	Петр	92 000	27.02.1980	313
2937	Сергеева	Елена	110 000	05.05.1982	310
2938	Катков	Георгий	112 000	16.12.1991	315



Пример:

- 1) Добавить сотрудника Кузнецова Сергея (id=3000, зарплата 100.000, дата рождения 31.07.1995) **в отдел 320**
- 2) Добавить сотрудника Кузнецова Сергея (id=3000, зарплата 100.000, дата рождения 31.07.1995) **в отдел 310**

department_id	department				
	employee_id	last_name	first_name	salary	birthday
310	2934	Иванов	Валерий	112 000	01.01.1966
	2935	Петров	Сергей	144 000	13.06.1977
	2937	Сергеева	Елена	110 000	05.05.1982
313	2936	Орлов	Петр	92 000	27.02.1980
315	2938	Катков	Георгий	112 000	16.12.1991





Целостность данных (data integrity) – соответствие значений всех данных базы данных определенному непротиворечивому набору правил.

Три класса правил поддержки целостности данных в реляционной БД:

- 1) Целостность доменов
- 2) Целостность сущностей
- 3) Целостность ссылок



Целостность доменов поддерживается за счет механизма доменных ограничений.

NULL-значение – маркер, используемый, чтобы обойти проблему неполных или неизвестных данных.

При работе с данными, которые могут содержать *NULL*-значения, необходимо использовать **трехзначную логику**.

Таблица истинности AND

AND	False	True	NULL
False	False	False	False
True	False	True	NULL
NULL	False	NULL	NULL

Таблица истинности OR

OR	False	True	NULL
False	False	True	NULL
True	True	True	True
NULL	NULL	True	NULL

Таблица истинности NOT

Not	
False	True
True	False
NULL	NULL



Имеется несколько следствий применения трехзначной логики:

- NULL-значение не равно самому себе. Действительно, выражение $NULL = NULL$ дает значение не ИСТИНА (*True*), а НЕИЗВЕСТНО (*NULL*). Соответственно выражение $x = x$ не обязательно ИСТИНА.
- Неверно также, что *NULL*-значение не равно самому себе. Выражение $NULL \neq NULL$ также принимает значение не ИСТИНА, а НЕИЗВЕСТНО. Соответственно выражение $x \neq x$ не обязательно ЛОЖЬ.
- Выражение $x \text{ OR } NOT(x)$ не обязательно ИСТИНА. Соответственно в трехзначной логике не работает принцип исключенного третьего (любое высказывание либо истинно, либо ложно).





Потенциальный ключ – подмножество атрибутов отношения, удовлетворяющее требованиям уникальности и минимальности (несократимости).

Уникальность означает, что не существует двух кортежей данного отношения, в которых значения этого подмножества атрибутов совпадают (равны).

Минимальность (несократимость) означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности.

Первичный ключ (primary key) – один из потенциальных ключей отношения, выбранный в качестве основного ключа. Если в отношении имеется единственный потенциальный ключ, он является и первичным ключом. Если потенциальных ключей несколько, один из них выбирается в качестве первичного, а другие называют «альтернативными».



Целостность сущностей направлена на обеспечение внутреннего единства отдельной сущности.

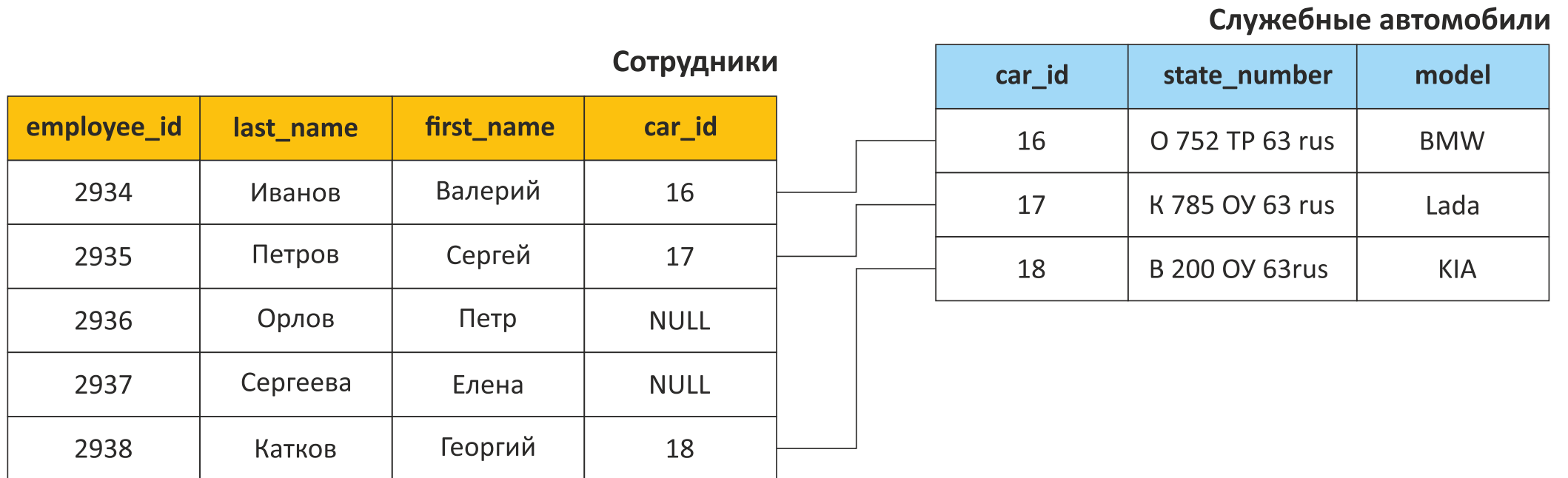
Каждая строка таблицы обязана быть уникальной. Для этого нужно контролировать корректность первичного ключа отношения. Суть требования к первичному ключу проста – во входящих в его состав атрибутах не должно содержаться ни одного определителя NULL.

Ни один из столбцов, входящих в состав первичного ключа, не должен допускать ввода определителя NULL



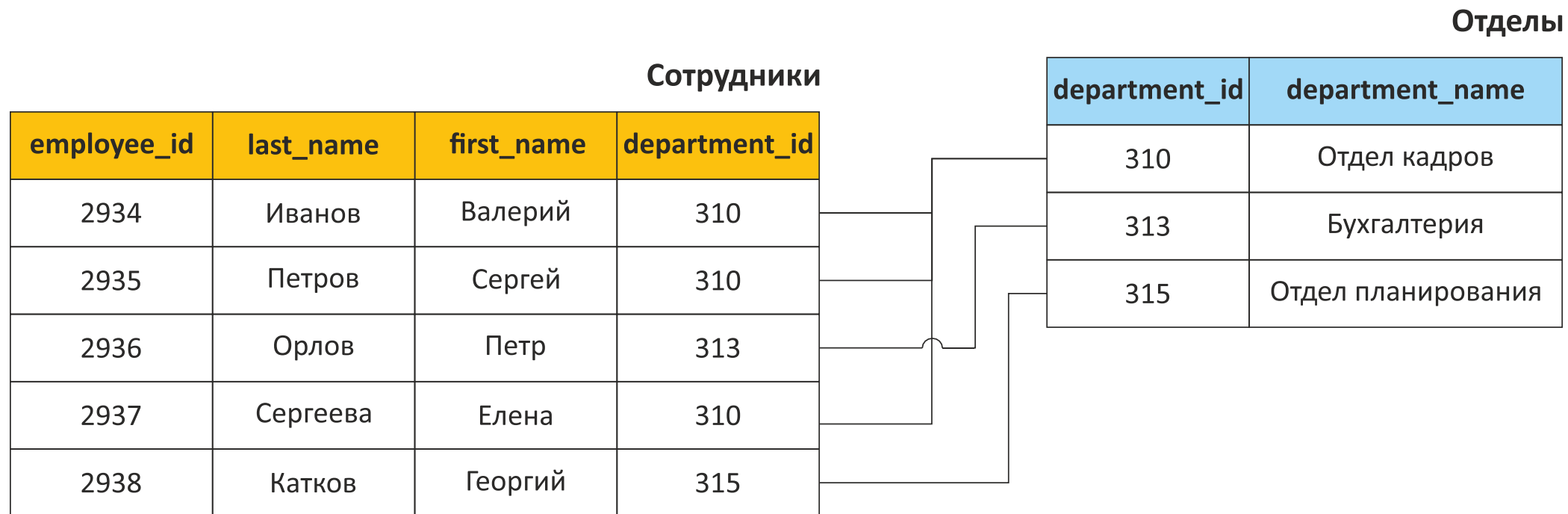
Связи между кортежами отношений реализуются через механизм **«внешних ключей»**, являющихся, по существу, ссылками на атрибуты связываемых кортежей отношений.

Типы связей: «один к одному», «один ко многим» и «многие ко многим».





Связь «один ко многим»





Связь «многие ко многим»

film

film_id	title	description
①	ACADEMY DINOSAUR	A Epic Drama of...
②	ACE GOLDFINGER	A Astounding Epistle of ...
3	ADAPTATION HOLES	A Astounding Reflection of ...
4	AFFAIR PREJUDICE	A Fanciful Documentary of ...
...

film_actor

actor_id	film_id
①	①
1	23
...	...
19	②
85	②
...	...

actor

actor_id	first_name	last_name
①	PENELOPE	GUINNESS
2	NICK	WAHLBERG
3	ED	CHASE
...



- Внешний ключ может быть простым и составным.
- Внешний ключ должен быть определен на тех же доменах, что и соответствующий первичный ключ родительского отношения.
- Внешний ключ, как правило, не обладает свойством уникальности.
- Хотя каждое значение внешнего ключа обязано совпадать со значениями потенциального ключа в некотором кортеже родительского отношения, то обратное, вообще говоря, неверно.
- Для внешнего ключа не требуется, чтобы он был компонентом некоторого потенциального ключа.
- NULL-значения для атрибутов внешнего ключа допустимы только в том случае, когда атрибуты внешнего ключа не входят в состав никакого потенциального ключа



Правило целостности внешних ключей: внешние ключи не должны быть несогласованными, т.е. для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительском отношении.

Ссылочная целостность может нарушиться в результате операций, изменяющих состояние базы данных: вставка, обновление и удаление кортежей в отношениях.

Т.к. в определении ссылочной целостности участвуют два отношения - родительское и дочернее, а в каждом из них возможны три операции - вставка, обновление, удаление, то нужно рассмотреть шесть различных вариантов.





Для родительского отношения:

- Вставка кортежа в родительском отношении. При вставке кортежа в родительское отношение возникает новое значение потенциального ключа. Т.к. допустимо существование кортежей в родительском отношении, на которые нет ссылок из дочернего отношения, то вставка кортежей в родительское отношение **не нарушает ссылочной целостности**.
- Обновление кортежа в родительском отношении. При обновлении кортежа в родительском отношении может измениться значение потенциального ключа. Если есть кортежи в дочернем отношении, ссылающиеся на обновляемый кортеж, то значения их внешних ключей станут некорректными. Обновление кортежа в родительском отношении **может привести к нарушению ссылочной целостности**, если это обновление затрагивает значение потенциального ключа.
- Удаление кортежа в родительском отношении. При удалении кортежа в родительском отношении удаляется значение потенциального ключа. Если есть кортежи в дочернем отношении, ссылающиеся на удаляемый кортеж, то значения их внешних ключей станут некорректными. Удаление кортежей в родительском отношении **может привести к нарушению ссылочной целостности**.





Для дочернего отношения:

- Вставка кортежа в дочернее отношение. Нельзя вставить кортеж в дочернее отношение, если вставляемое значение внешнего ключа некорректно. Вставка кортежа в дочернее отношение **может привести к нарушению ссылочной целостности**.
- Обновление кортежа в дочернем отношении. При обновлении кортежа в дочернем отношении можно попытаться некорректно изменить значение внешнего ключа. Обновление кортежа в дочернем отношении **может привести к нарушению ссылочной целостности**.
- Удаление кортежа в дочернем отношении. При удалении кортежа в дочернем отношении **ссылочная целостность не нарушается**.



Основные стратегии поддержания ссылочной целостности:

- RESTRICT (ОГРАНИЧИТЬ) – не разрешать выполнение операции, приводящей к нарушению ссылочной целостности. Это самая простая стратегия, требующая только проверки, имеются ли кортежи в дочернем отношении, связанные с некоторым кортежем в родительском отношении.
- CASCADE (ВЫПОЛНИТЬ КАСКАДНО) – разрешить выполнение требуемой операции, но внести при этом необходимые поправки в других отношениях так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи. Изменение начинается в родительском отношении и каскадно выполняется в дочернем отношении.



Дополнительные стратегии поддержания ссылочной целостности:

- SET NULL (УСТАНОВИТЬ В NULL) – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на NULL-значения. Эта стратегия имеет два недостатка. Во-первых, для нее требуется допустить использование NULL-значений. Во-вторых, кортежи дочернего отношения теряют всякую связь с кортежами родительского отношения. Установить, с каким кортежем родительского отношения были связаны измененные кортежи дочернего отношения, после выполнения операции уже нельзя.
- SET DEFAULT (УСТАНОВИТЬ ПО УМОЛЧАНИЮ) – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на некоторое значение, принятое по умолчанию.
- IGNORE (ИГНОРИРОВАТЬ) – выполнять операции, не обращая внимания на нарушения ссылочной целостности. В этом случае в дочернем отношении могут появляться некорректные значения внешних ключей, и вся ответственность за целостность базы данных ложится на пользователя.



Реляционная алгебра представляет собой теоретический язык операций, которые на основе одного или нескольких отношений позволяют создавать другое отношение.

Операции реляционной алгебры:

- 1) выборка (selection) σ ;
- 2) проекция (projection) Π ;
- 3) декартово произведение (cartesian product) \times ;
- 4) объединение, сложение (union) \cup ;
- 5) вычитание, разность (set difference) $-$;
- 6) пересечение (intersection) \cap ;
- 7) деление (division) \div ;
- 8) соединение (join) \Join .



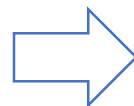


$$R' = \sigma_F(R),$$

где F – предикат выборки, R – множество кортежей, R' – подмножество кортежей, для которых предикат выборки F истинен.

R – таблица «Студенты»

Id	last_name	first_name	birthday	speciality
1	Орехов	Владимир	11.04.1999	Математика
2	Петровский	Александр	22.11.2000	Математика
3	Кульгина	Оксана	05.01.2000	Ин. язык
4	Самойлов	Евгений	15.07.2001	Информатика
5	Кузьмина	Ирина	02.03.2000	Физика
6	Яковенко	Константин	13.09.2001	Химия
7	Ищенко	Владимир	19.09.2002	Астрономия



R' – выборка из таблицы «Студенты»

2	Петровский	Александр	22.11.2000	Математика
3	Кульгина	Оксана	05.01.2000	Ин. язык
5	Кузьмина	Ирина	02.03.2000	Физика



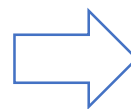
Проекция

$$R' = \prod_{i_1, i_2, \dots, i_k} (R),$$

где R – множество кортежей, n – количество атрибутов множества R , i_1, \dots, i_k – подмножество атрибутов для проекции, $k \leq n$.

R – таблица «Студенты»

Id	last_name	first_name	birthday	speciality
1	Орехов	Владимир	11.04.1999	Математика
2	Петровский	Александр	22.11.2000	Математика
3	Кульгина	Оксана	05.01.2000	Ин. язык
4	Самойлов	Евгений	15.07.2001	Информатика
5	Кузьмина	Ирина	02.03.2000	Физика
6	Яковенко	Константин	13.09.2001	Химия
7	Ищенко	Владимир	19.09.2002	Астрономия



Проекция

Орехов	Владимир
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Яковенко	Константин
Ищенко	Владимир





Декартово произведение

$$R' = R \times S$$

Кортежи результирующего отношения R' формируются путем попарного соединения (конкатенации, или сцепления) всех кортежей отношений-операндов R и S .

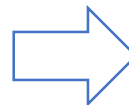
R – «Студенты»

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана

×

S – «Дисциплины»

discipline
Операционные системы
Языки программирования
Методы программирования
СУБД



$R \times S$

last_name	first_name	discipline
Орехов	Владимир	Операционные системы
Орехов	Владимир	Языки программирования
Орехов	Владимир	Методы программирования
Орехов	Владимир	СУБД
Петровский	Александр	Операционные системы
Петровский	Александр	Языки программирования
Петровский	Александр	Методы программирования
Петровский	Александр	СУБД
Кульгина	Оксана	Операционные системы
Кульгина	Оксана	Языки программирования
Кульгина	Оксана	Методы программирования
Кульгина	Оксана	СУБД





Объединение

$$R' = R \cup S$$

Результирующее отношение R' включает все кортежи, входящие хотя бы в одно из отношений-операндов R или S .

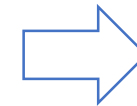
R – «Студенты 1»

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана

\cup

S – «Студенты 2»

last_name	first_name
Алфёров	Николай
Остапенко	Владимир
Юров	Олег
Орехов	Владимир



$R \cup S$

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана
Алфёров	Николай
Остапенко	Владимир
Юров	Олег



Вычитание

$$R' = R - S$$

Результирующее отношение включает все кортежи, входящие в отношение-операнд R , такие, что ни один из них не входит в отношение-операнд S .

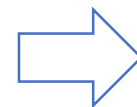
R – «Студенты 1»

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Яковенко	Константин
Ищенко	Владимир
Алфёров	Николай

—

S – «Студенты 2»

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Ищенко	Владимир
Алфёров	Николай



$R - S$

last_name	first_name
Яковенко	Константин



Пересечение

$$R' = R \cap S$$

Результирующее отношение R' включает все кортежи, входящие в оба отношения-операнд R или S .

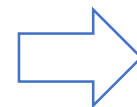
R – «Студенты 1»

last_name	first_name
Орехов	Владимир
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Яковенко	Константин
Щукин	Александр
Алфёров	Николай

\cap

S – «Студенты 2»

last_name	first_name
Бобров	Семен
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Ищенко	Владимир
Алфёров	Николай



$R \cap S$

last_name	first_name
Петровский	Александр
Кульгина	Оксана
Самойлов	Евгений
Кузьмина	Ирина
Алфёров	Николай





Деление

$$R' = R/S$$

- Результат деления исходной таблицы R на таблицу-делитель S будет содержать столбцы, отсутствующие в делителе
- В качестве строк в результат войдут только те записи из делителя S , что при декартовом произведении результата на делитель содержатся в делимом R .

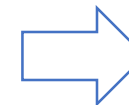
R – делимое

last_name	discipline	type
Орлов	СУБД	Лекция
Орлов	СУБД	Лаб. работа
Орлов	Языки программирования	Практика
Володина	СУБД	Лаб. работа
Володина	Языки программирования	Практика
Семенов	Языки программирования	Практика

/

S – делитель

discipline	type
СУБД	Лаб. работа
Языки программирования	Практика



R/S

last_name
Орлов
Володина



Соединение

$$R' = R \bowtie S$$

Кортежи результирующего отношения R' образуются путем соединения (конкатенации, или сцепления) кортежей отношений-операндов R и S , удовлетворяющих заданному условию (любому корректному логическому выражению).

Выполнение операции соединения отношений можно рассматривать как операцию их расширенного декартова произведения с последующей фильтрацией множества кортежей полученного промежуточного отношения по заданному *условию*.

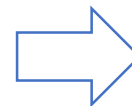
R – «Преподаватели»

last_name	chair_id
Орлов	1
Володина	1
Семенов	3
Архипов	2
Андреев	2



S – «Кафедры»

chair_id	chair
1	Геоинформатика
2	Высшая математика
3	Физика
4	Химия



R' – естественное соединение

last_name	chair_id	chair
Орлов	1	Геоинформатика
Володина	1	Геоинформатика
Семенов	3	Физика
Архипов	2	Высшая математика
Андреев	2	Высшая математика





Соединение

R – «Преподаватели»

last_name	chair_id
Орлов	1
Володина	1
Семенов	3
Архипов	2
Андреев	2

$\triangleright \cup$

S – «Кафедры»

chair_id	chair
1	Геоинформатика
2	Высшая математика
3	Физика
4	Химия



R' – правое внешнее соединение

last_name	chair_id	chair
Орлов	1	Геоинформатика
Володина	1	Геоинформатика
Семенов	3	Физика
Архипов	2	Высшая математика
Андреев	2	Высшая математика
NULL	4	Химия





- **Внутреннее соединение** (inner join) — соединяются только те кортежи отношений-операндов, для которых выполняется заданное условие
- **Левое** (left join) и **правое** (right join) **внешнее соединение** — результирующее отношение будет безусловно содержать все кортежи левого (или соответственно правого) отношения-операнда, в том числе и те, для которых нет «пары» в другом отношении-операнде, при этом «недостающие» атрибуты в таких кортежах результирующего отношения получают неопределенные NULL-значения
- **Внешнее соединение** (outer join) — одновременно и левое, и правое соединение
- **Экви-соединение** (equal join) — такое соединение, условие которого содержит оператор сравнения «равно»
- **Естественное соединение** (natural join) — экви-соединение двух отношений, имеющих одинаковые атрибуты (как правило, это первичный и внешний ключи соединяемых отношений), равенство которых и является условием соединения кортежей (при этом совпадающий атрибут в схеме результирующего отношения не дублируется)



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

**БЛАГОДАРЮ
ЗА ВНИМАНИЕ**

Агафонов А.А.
д.т.н., доцент кафедры ГИИБ