Севастопольский государственный университет
Кафедра «Информационные системы»

Управление данными

курс лекций

лектор:
ст. преподаватель кафедры ИС Абрамович А.Ю.

Лекция 1

ОБЗОРНАЯ ЛЕКЦИЯ

Основные понятия БД. Модели данных.

Реляционная Алгебра

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Банк данных – это система специальным образом организованных данных – баз данных, а также технических, программных, языковых и организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

База данных (БД) – совокупность данных, организованных по определённым правилам, которые предусматривают общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

База данных (БД) – поименованная совокупность взаимосвязанных данных, находящихся под управлением СУБД.

База данных (БД) – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

СУБД – совокупность языковых и программных средств, предназначенных для создания, ведения и использования информации, хранящейся в БД.

ОСНОВНЫЕ ФУНКЦИИ СУБД

Создание баз данных, изменение, удаление и объединение их по определённым признакам.

Хранение данных, в том числе больших массивов, в структурированном виде и нужном формате

Защита данных от взлома и нежелательных изменений при помощи распределённого доступа: когда разным группам пользователей доступны разный объём и сегменты данных

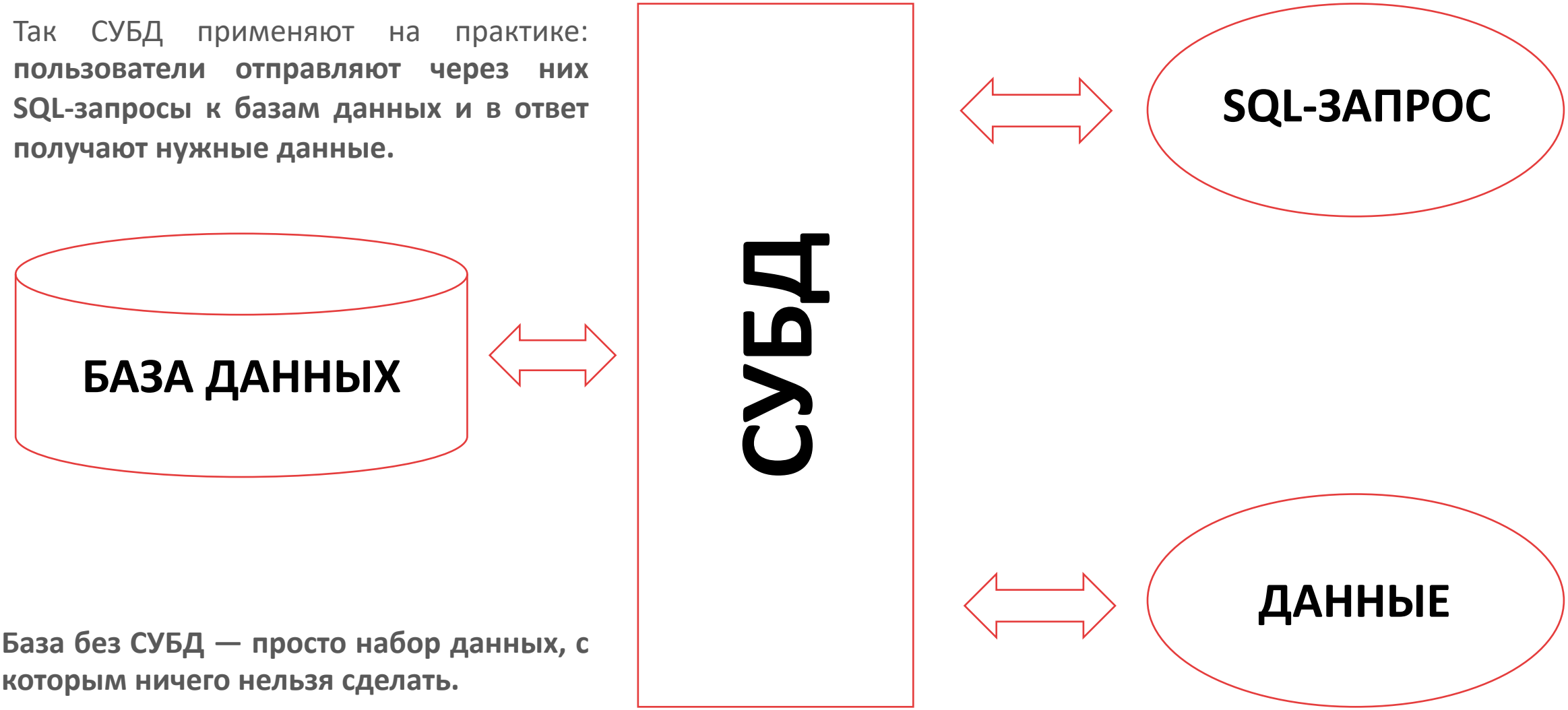
Выгрузка и сортировка данных по заданным фильтрам при помощи SQL-запросов

Поддержка целостности баз данных, резервное копирование и восстановление после сбоёв

Основная функция СУБД – это **предоставление пользователю БД возможности работы с ней**, не вникая в детали на уровне аппаратного обеспечения. **То есть все запросы пользователя к БД, добавление и удаление данных, выборки, обновление данных – все это обеспечивает СУБД.**

КОМПОНЕНТЫ СУБД

Так СУБД применяют на практике: **пользователи отправляют через них SQL-запросы к базам данных и в ответ получают нужные данные.**



База без СУБД — просто набор данных, с которым ничего нельзя сделать.

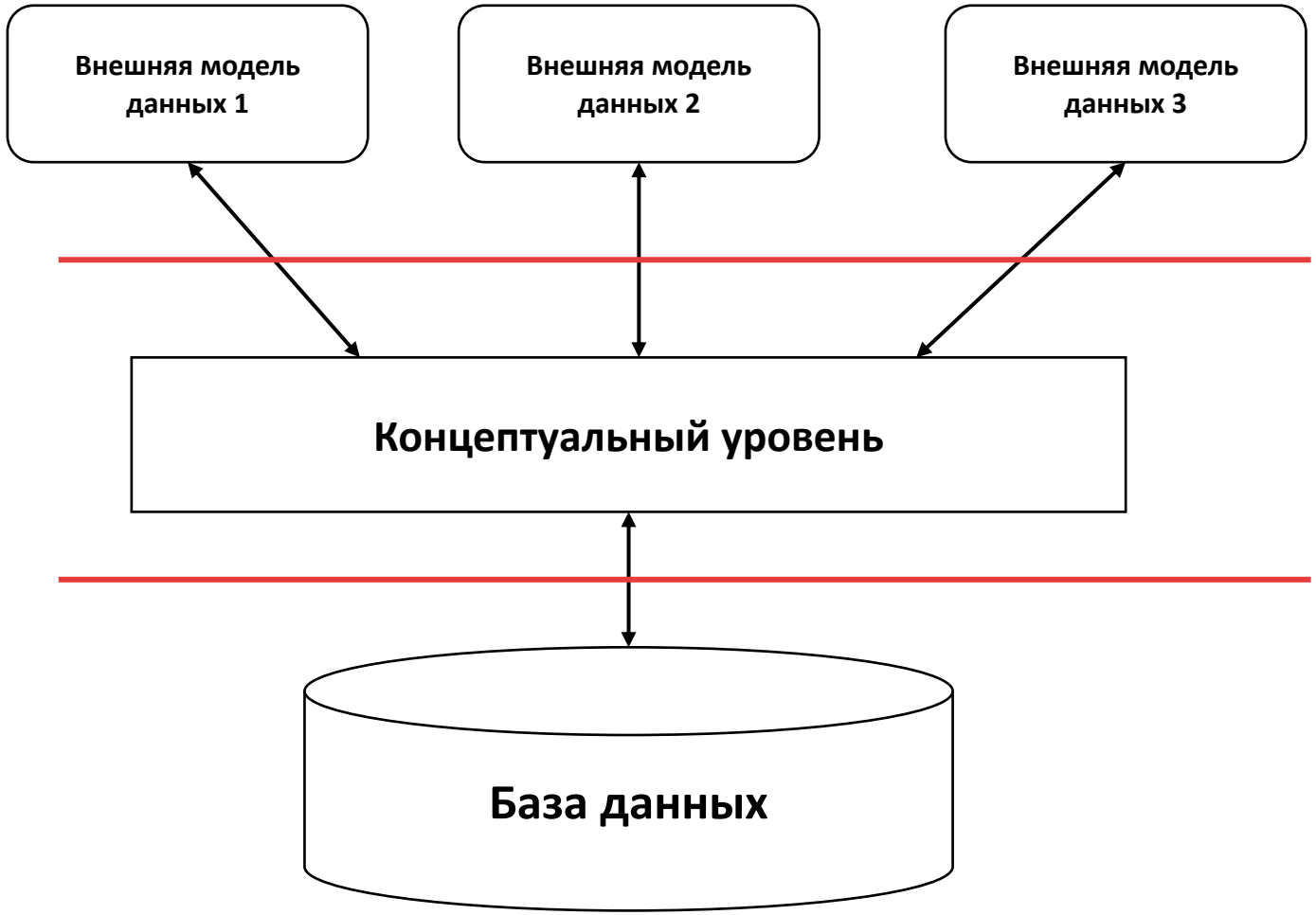
АРХИТЕКТУРА БАЗЫ ДАННЫХ

Трехуровневая система организации БД

Уровень внешних моделей — является самым верхним уровнем или уровнем пользователя. Это совокупность внешних представлений данных, которые обрабатывают приложения и какими их видит пользователь на экране.

Концептуальный уровень — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, объединяет данные. Фактически отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась БД.

Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.



Одними из основополагающих в концепции баз данных являются обобщенные категории *«данные»* и *«модель данных»*.

«Иванов Иван Иванович 24»

«Управление данными 36 36»

«Печенье Темная сторона 30 80»

Данные не обладают определенной структурой, **данные становятся информацией тогда, когда пользователь задает им определенную структуру**, то есть осознаёт их смысловое содержание. Поэтому **центральным понятием в области баз данных является понятие модели данных.**

ФИО студента	Осталось лабораторных сдать, шт.
Иванов Иван Иванович	24

Дисциплина	Лекции, ч.	ЛЗ, ч.
Управление данными	36	36

Товар	Поставщик	Количество, коробки шт.	Цена за ед., руб.
Печенье	Темная сторона	30	80

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Модель данных - интегрированный набор понятий для описания и обработки данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Модель данных - схема (порядок, совокупность принципов, система) организации данных в единое целое для создания, накопления, обработки и управления. Это **некоторая абстракция**, которая будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию.

Модель данных - совокупность структур данных и операций по их обработке.

Структурная часть

- набор правил, по которым может быть построена база данных

Управляющая часть

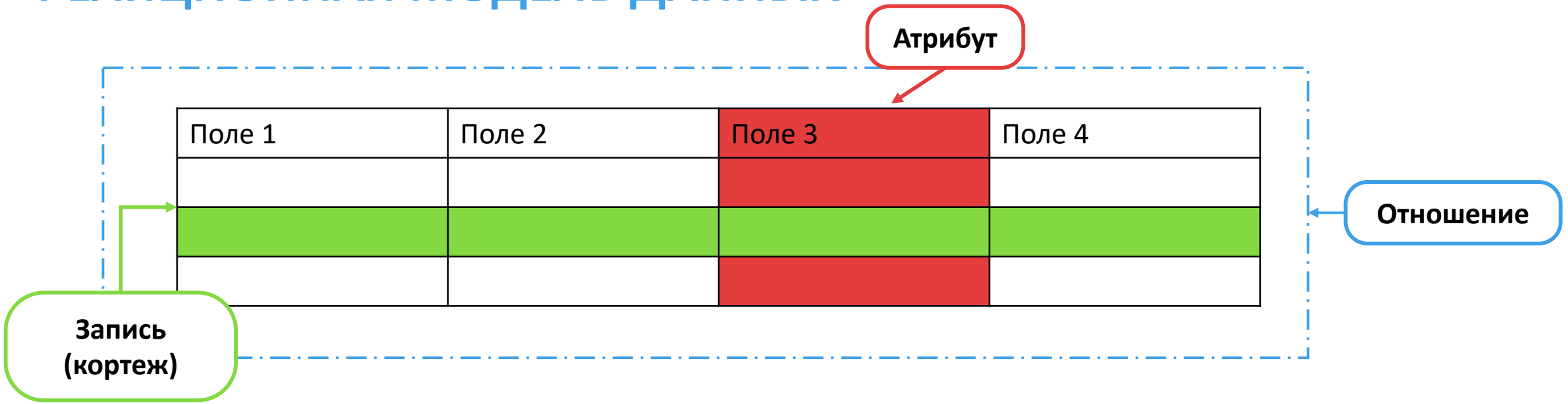
- определяющая типы допустимых операций с данными

Ограничения

- набор (необязательный) ограничений поддержки целостности данных, гарантирующих корректность используемых данных.

КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ



Реляционная модель данных предложена сотрудником фирмы IBM Эдгаром Коддом и **основывается на понятии отношение (relation).**

Реляционная модель данных представляет собой **совокупность данных, содержащихся в двухмерных таблицах,** соединённых между собой отношениями. Любые данные можно преобразовать в простую таблицу. **Такое представление является наиболее удобным и для пользователя, для и машины.**

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

РЕЛЯЦИОННАЯ МОДЕЛЬ СОСТОИТ ИЗ ТРЕХ ЧАСТЕЙ:

Структурная часть	• описывает, какие объекты рассматриваются реляционной моделью.
Целостная часть	• описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. **Это целостность сущностей и целостность внешних ключей.**
Манипуляционная часть	• описывает два эквивалентных способа манипулирования реляционными данными - **реляционную алгебру и реляционное исчисление**.

Единственные **структуры данных, используемые в реляционной модели**, являются нормализованные **n-арные отношения**.

РЕЛЯЦИОННАЯ АЛГЕБРА ИЛИ РЕЛЯЦИОННОЕ ИСЧИСЛЕНИЕ

Выражения реляционной алгебры строятся на основе алгебраических операций (высокого уровня), и подобно тому, как интерпретируются арифметические и логические выражения, выражение реляционной алгебры также имеет процедурную интерпретацию.

Теоретико-множественные операции	Специальные реляционные операции
• объединение - ∪ • вычитание (разность) • пересечение - ∩ • декартово произведение - ×	• проекция - π • выборка (селекция) - σ • соединение - ⋈ • деление - /

Операции объединения, пересечения и разности требуют от операндов совместимости по типу.

Два отношения совместимы по типу, если:

1. каждое из них имеет одно и то же множество имен атрибутов (одна и та же степень),

2. соответствующие атрибуты (с одинаковыми именами) определены на одном и том же домене.

Проекция (project)

Проекция является операцией, при которой из отношения выделяются атрибуты только из указанных доменов, то есть **из таблицы выбираются только нужные столбцы**, при этом, если получится несколько одинаковых кортежей, то в результирующем отношении остается только по одному экземпляру подобного кортежа.

Отношение A

COMPANY	PRODUCT	DRIVERS
ООО «Темная сторона»	Печеньки	Петр
ООО «Темная сторона»	Чай	Петр
ОАО «Овощи»	Огурцы	Олег
ООО «Молочко»	Йогурт	Иван

Проекция $\pi_{COMPANY,DRIVERS}(A)$

COMPANY	DRIVERS
ООО «Темная сторона»	Петр
ОАО «Овощи»	Олег
ООО «Молочко»	Иван

```
SELECT DISTINCT COMPANY,DRIVERS FROM A;
```

В SQL **для полного соответствия операции** проекции необходимо **указывать ключевое слово DISTINCT.**

Селекция (select)

Селекция — это операция, которая **выделяет множество строк в таблице, удовлетворяющих заданным условиям.** Условием может быть любое логическое выражение.

Отношение A

COMPANY	DRIVERS
ООО «Темная сторона»	Петр
ОАО «Овощи»	Петр
ООО «Молочко»	Иван

Селекция $\sigma_{DRIVERS='Петр'}(A)$

COMPANY	DRIVERS
ООО «Темная сторона»	Петр
ОАО «Овощи»	Петр

```
SELECT * FROM A WHERE DRIVERS='Петр';
```

Соединение (JOIN)

Эта операция определяет **подмножество декартова произведения двух разносхемных отношений.** Кортеж декартова произведения входит в результирующее отношение, если для атрибутов разных исходных отношений выполняется некоторое условие соединения ***F***. Соединение может быть выражено следующим образом:

$$R \triangleright\!\triangleleft_F S = \sigma_F (R \times S).$$

Соединение имеет две разновидности: **естественное соединение и соединение по условию (θ - соединение).**

Естественным соединением отношений A(X,Y) и B(Y,T) (A JOIN B) называется отношение с заголовком {X, Y, T} и с телом, содержащим множество всех кортежей вида <X:x, Y:y, T:t> таких, для которых в отношении A значение атрибута X равно x, а значение атрибута Y равно y, и в отношении B значение атрибута Y равно y, а атрибута T равно t. **При естественном соединении производится сцепление строк операндов соединения по общим атрибутам.**

SELLERS

ID	COMPANY
234	ООО «Темная сторона»
124	ОАО «Овощи»
188	ООО «Молочко»

PRODUCTS

ID	NAME	COMPANY	PRICE
23	Печеньки	ООО «Темная сторона»	90
12	Молоко	ООО «Молочко»	220
18	Огурцы	ОАО «Овощи»	108

SELLERS ▷◁ PRODUCTS

SELLERS.ID	COMPANY	PRODUCTS.ID	NAME	COMPANY	PRICE
234	ООО «Темная сторона»	23	Печеньки	ООО «Темная сторона»	90
124	ОАО «Овощи»	18	Огурцы	ОАО «Овощи»	108
188	ООО «Молочко»	12	Молоко	ООО «Молочко»	220

Замечание 1. Соединения не всегда выполняются по внешнему ключу и соответствующему потенциальному ключу, хотя такие соединения очень распространены и являются важным частным случаем.

Замечание 2. Если отношения **A и B** не имеют общих атрибутов, то выражение **A JOIN B** эквивалентно **A ×B**.

θ - соединение – это отношение с тем же заголовком, что и при декартовом произведении отношений **A** и **B**, и с телом, содержащим множество кортежей **t ∈ A×B**, таких что вычисление условия **X ▷◁ Y** дает значение истина для данного кортежа.

Атрибуты X и Y должны быть определены на одном и том же домене, а оператор должен иметь смысл для этого домена.

SELLERS

ID	COMPANY
234	ООО «Темная сторона»
124	ОАО «Овощи»
188	ООО «Молочко»

PRODUCTS

ID	NAME	PRICE
23	Печеньки	90
12	Молоко	220
18	Огурцы	108

SELLERS ▷◁ PRODUCTS $\sigma_{\text{PRICE>100}}$**(*Products*)**

SELLERS.ID	COMPANY	PRODUCTS.ID	NAME	COMPANY	PRICE
124	ОАО «Овощи»	18	Огурцы	ОАО «Овощи»	108
188	ООО «Молочко»	12	Молоко	ООО «Молочко»	220

Операция θ-соединение эквивалентна двум операциям: нахождению расширенного декартова произведения двух отношений (при необходимости с переименованием соответствующих атрибутов) и последующему выполнению указанной выборки из полученного результата.

Дана реляционная база данных:

Рейс (№_рейса, пункт_отправления, пункт_назначения, время_вылета, стоимость);
Полет (дата полета, № рейса, код_экипажа, свободные_места, тип_самолета, объем_груза);
Самолет (тип_самолета, число_экипажа, количество_мест, вес_груза);

Форма представления запроса на РА:

$$R = \pi_{\text{столбцы, которые выбираются}}(\text{наименование отношения})$$

Определить число свободных мест по всем рейсам на 20.06.23.

$$R = \pi\ \text{№рейса, свободные\_места}\ (\sigma\text{дата} = 20.06.23\ (\ \text{Полет}\ ))$$

Определить рейсы и время вылета из Симферополя в Москву.

$$R = \pi\ \text{№рейса, время\_вылета}\ (\sigma\ \text{пункт\_отправления} = \text{« Симферополь »} \wedge \text{пункт\_назначения} = \text{« Москва »}\ (\ \text{Рейс}\ ))$$

Определить даты рейсов Москва – Калининград.

$$R = \pi\ \text{дата, номер\_рейса}\ (\sigma\ \text{пункт\_отправления} = \text{« Москва »} \wedge \text{пункт\_назначения} = \text{«Калининград»}\ (\ \text{Рейс} \rhd\lhd \text{Полет}\ )\ )$$

Объединение (UNION)

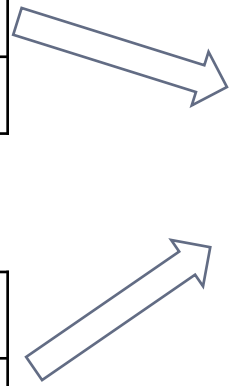
Объединением двух совместимых по типу отношений **A и B (A ∪ B)** называется отношение с тем же заголовком, как в отношениях **A** и **B**, и с телом, состоящим из множества кортежей **t**, принадлежащих **A** или **B** или обоим отношениям.

A

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1996	Иванов	Рязань	23
1777	Сидоров	Тверь	42

B

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1744	Липко	Сочи	22



A ∪ B

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1996	Иванов	Рязань	23
1777	Сидоров	Тверь	42
1744	Липко	Сочи	22

Предложение **UNION** имеет следующую **общую форму**:

```
оператор_SELECT UNION [ ALL | DISTINCT ] оператор_SELECT
```

оператор_SELECT — это **любой подзапрос SELECT** без предложений **ORDER BY, LIMIT, FOR NO KEY UPDATE, FOR UPDATE, FOR SHARE и FOR KEY SHARE.**

Оператор UNION вычисляет объединение множеств всех строк, возвращённых заданными запросами SELECT. Два оператора SELECT должны выдавать **одинаковое число столбцов, а типы соответствующих столбцов должны быть совместимыми.**

Результат UNION **не будет содержать повторяющихся строк**, если **не указан параметр ALL**. **ALL предотвращает исключение дубликатов.** DISTINCT можно записать явно, чтобы обозначить, что дублирующиеся строки должны удаляться (по умолчанию).

При использовании в одном запросе SELECT **нескольких операторов UNION** они вычисляются **слева направо, если иной порядок не определяется скобками**.

Пересечение (intersect)

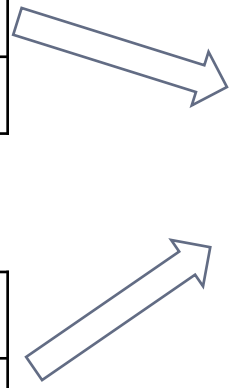
Пересечением двух совместимых по типу отношений **A и B (A ∩ B)** называется отношение с тем же заголовком, как в отношениях **A** и **B**, и с телом, состоящим из множества кортежей **t**, принадлежащих одновременно обоим отношениям **A** и **B**.

A

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1996	Иванов	Рязань	23
1777	Сидоров	Тверь	42

B

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1744	Липко	Сочи	22



A ∩ B

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43

Создает отношение, включающее все кортежи, входящие в оба отношения-операнда.

Предложение **INTERSECT** имеет следующую **общую форму**:

```
оператор_SELECT INTERSECT [ ALL | DISTINCT ] оператор_SELECT
```

оператор_SELECT — это **любой подзапрос SELECT** без предложений **ORDER BY, LIMIT, FOR NO KEY UPDATE, FOR UPDATE, FOR SHARE и FOR KEY SHARE.**

Оператор INTERSECT вычисляет пересечение множеств всех строк, возвращённых заданными запросами SELECT.

Результат INTERSECT **не будет содержать повторяющихся строк, если не указан параметр ALL.** С параметром ALL строка, повторяющаяся **m раз в левой таблице и n раз в правой**, будет выдана в результирующем наборе **min(m,n)** раз. DISTINCT можно записать явно, чтобы обозначить, что дублирующиеся строки должны удаляться (по умолчанию).

При использовании в одном запросе SELECT **нескольких операторов INTERSECT** они вычисляются **слева направо, если иной порядок не диктуется скобками**. INTERSECT связывает свои подзапросы сильнее, чем UNION. **A UNION B INTERSECT C будет восприниматься как A UNION (B INTERSECT C).**

Разность (except)

Разностью двух совместимых по типу отношений **A и B (A – B)** называется отношение с тем же заголовком, как в отношениях **A и B**, и с телом, состоящим из множества кортежей **t**, принадлежащих отношению **A** и не принадлежащих отношению **B**.

A

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1996	Иванов	Рязань	23
1777	Сидоров	Тверь	42

B

ID_NUM	NAME	CITY	AGE
1809	Петров	Москва	43
1744	Липко	Сочи	22

A - B

ID_NUM	NAME	CITY	AGE
1996	Иванов	Рязань	23
1777	Сидоров	Тверь	42

Все кортежи, входящие в первое отношение, такие, что ни один из них не входит во второе отношение.

Предложение **EXCEPT** имеет следующую **общую форму**:

```
оператор_SELECT EXCEPT [ ALL | DISTINCT ] оператор_SELECT
```

оператор_SELECT — это **любой подзапрос SELECT** без предложений **ORDER BY, LIMIT, FOR NO KEY UPDATE, FOR UPDATE, FOR SHARE и FOR KEY SHARE.**

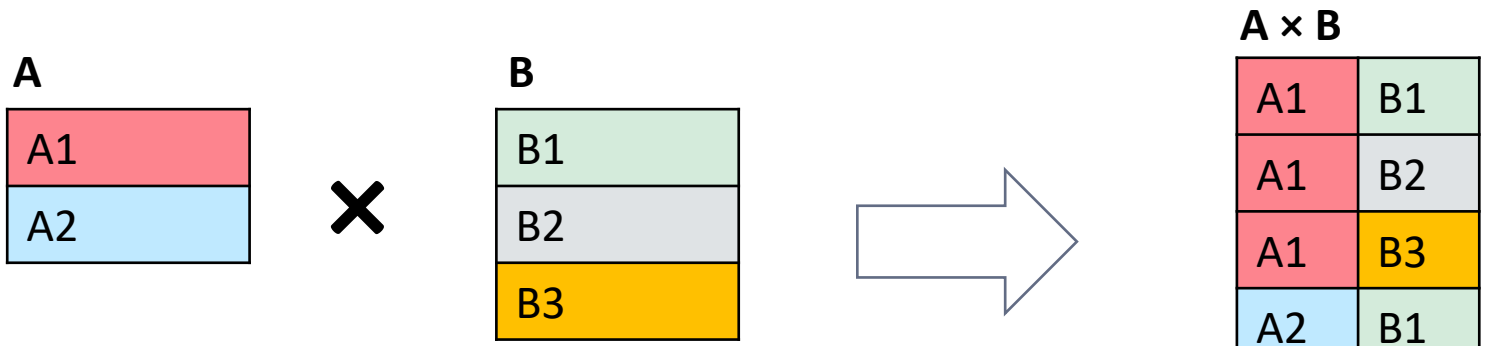
Оператор EXCEPT вычисляет набор строк, которые присутствуют в результате левого запроса SELECT, но отсутствуют в результате правого.

Результат EXCEPT **не будет содержать повторяющихся строк**, если **не указан параметр ALL**. С параметром ALL строка, повторяющаяся **m раз в левой таблице и n раз в правой**, будет выдана в результирующем наборе **max(m-n,0) раз**. DISTINCT можно записать явно, чтобы обозначить, что дублирующиеся строки должны удаляться (по умолчанию).

При использовании в одном запросе SELECT **нескольких операторов EXCEPT** они вычисляются **слева направо, если иной порядок не диктуется скобками**. EXCEPT связывает свои подзапросы так же сильно, как UNION.

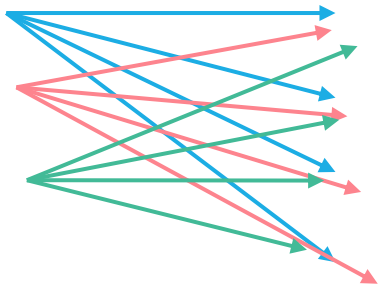
Декартово произведение (cartesian product)

Декартово произведение двух отношений **A и B (A × B)**, где **A** и **B** не имеют общих имен атрибутов, определяется как отношение с заголовком, представляющим собой сцепление двух заголовков исходных отношений **A** и **B**, и телом, состоящим из множества кортежей **t** таких что первым является любой кортеж отношения **A**, а вторым – любой кортеж, принадлежащий отношению **B**.



SELLERS

ID	COMPANY
234	ООО «Темная сторона»
124	ОАО «Овощи»
188	ООО «Молочко»



PRODUCTS

ID	NAME	PRICE
234	Печеньки	90
188	Молоко	220
140	Чай	120
124	Огурцы	108

SELLERS × PRODUCTS

SELLERS.ID	COMPANY	PRODUCTS.ID	NAME	PRICE
234	ООО «Темная сторона»	234	Печеньки	90
234	ООО «Темная сторона»	188	Молоко	220
188	ООО «Молочко»	234	Печеньки	90
188	ООО «Молочко»	188	Молоко	220

SELECT * FROM SELLERS, PRODUCT;

SELECT * FROM SELLERS CROSS JOIN PRODUCT;

Дана реляционная база данных:

Рейс (№_рейса, пункт_отправления, пункт_назначения, время_вылета, стоимость);

Полет (дата полета, № рейса, код_экипажа, свободные_места, тип_самолета, объем_груза);

Самолет (тип_самолета, число_экипажа, количество_мест, вес_груза);

Определить типы самолетов , которые использовались как в январе, так и в феврале 2023 года.

R1 = **π**тип_самолета (**σ**дата > =01.01.23 ∧ дата <=31.01.23(Полет));

R2 = **π** тип_самолета (**σ** дата >= 01.02.23 ∧ дата < 29.02.23(Полет));

R = R1 ∩ R2

Определить номера рейсов, которые не производились с даты А по дату Б.

R1 = **π**№_рейса (Рейс);

R2 = **π**№_рейса (**σ** дата >A ∧ дата <B(Полет));

R = R1 - R2;

Деление (division)

Делением отношений **A(X,Y)** на **B(Y)** **(A/B)** называется отношение с заголовком **{X}** и телом, содержащим множество всех кортежей **{X:x}**, таких что существует кортеж **{X:x, Y:y},** который принадлежит отношению **A** для всех кортежей **{Y:y}**, принадлежащих отношению **B**.

Отношение A

id_cartoon	name_cartoon	name_channel
0	The Simpsons	2x2
0	The Simpsons	Пятница
0	The Simpsons	Суббота
1	Family Guy	2x2
1	Family Guy	Пятница
2	Duck Tales	Суббота
2	Duck Tales	Пятница

Отношение B

name_channel
2x2
Пятница

Отношение A/B

id_cartoon	name_cartoon
0	The Simpsons
1	Family Guy

*Family Guy и The Simpsons — мультфильмы, которые показывались и на 2x2 и на Пятнице (условие во второй таблице). При этом Duck Tales не показывалось по 2x2, потому **был исключён из результирующей таблицы**.*

ОПЕРАЦИЯ ДЕЛЕНИЯ ПОЛЕЗНА ТОГДА, КОГДА ЗАПРОС СОДЕРЖИТ СЛОВО «ВСЕ».

не поддерживает операцию DIVIDE

```
SELECT DISTINCT c1.id_cartoon AS id, c1.name_cartoon AS
name_cartoon
        FROM CartoonsChanels AS c1
        WHERE NOT EXISTS
        (
            SELECT Chanels.name_channel FROM Chanels
            WHERE Chanels.name_channel NOT IN
            (
                SELECT c2.name_channel
                FROM CartoonsChanels AS c2
                WHERE c2.name_cartoon=c1.name_cartoon
            )
        );
```

Дана реляционная база данных:

Рейс (№_рейса, пункт_отправления, пункт_назначения, время_вылета, стоимость);

Полет (дата полета, № рейса, код_экипажа, свободные_места, тип_самолета, объем_груза);

Самолет (тип_самолета, число_экипажа, количество_мест, вес_груза);

Определить дату, когда осуществляют рейсы все возможные типы самолетов.

R1 = πдата_полета, тип_самолета (Полет);

R2 = π тип_самолета (Самолет);

R = R1/R2

Определить типы самолетов, осуществляющие рейсы из Москвы по всем возможным направлениям.

R1 = πтип_самолета, пункт_назначения ((σпункт_отправления = ‘Москва’ (Рейс ▷◁ Полет));

R2 = πпункт_назначения(Рейс);

R = R1/R2