

# 1. Запросы с использованием различных видов соединений таблиц (Inner join, Left join, Right join, Full join, Cross join, Cross apply, самосоединение).

## Inner join

```
SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
INNER JOIN City AS d ON d.ID=e.CityID
INNER JOIN Country AS p ON p.ID=d.CountryID
```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Турция	Анталья
3	3	2	Россия	Сочи
4	4	2	Россия	Сочи

## Left join

```
SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
LEFT JOIN City AS d ON d.ID=e.CityID
LEFT JOIN Country AS p ON p.ID=d.CountryID
```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Турция	Анталья
3	3	2	Россия	Сочи
4	4	2	Россия	Сочи

## Right join

```
SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
RIGHT JOIN City AS d ON d.ID=e.CityID
RIGHT JOIN Country AS p ON p.ID=d.CountryID
```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	3	2	Россия	Сочи
3	4	2	Россия	Сочи
4	NULL	NULL	Латвия	NULL
5	2	4	Турция	Анталья

## Full join

```
SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
FULL JOIN City AS d ON d.ID=e.CityID
FULL JOIN Country AS p ON p.ID=d.CountryID
```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Турция	Анталья
3	3	2	Россия	Сочи
4	4	2	Россия	Сочи
5	NULL	NULL	Латвия	NULL

## Cross join

```
SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
CROSS JOIN City AS d
CROSS JOIN Country AS p
```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Россия	Москва
3	3	2	Россия	Москва
4	4	2	Россия	Москва
5	1	1	Россия	Сочи
6	2	4	Россия	Сочи
7	3	2	Россия	Сочи
8	4	2	Россия	Сочи
9	1	1	Россия	Анталья
10	2	4	Россия	Анталья
11	3	2	Россия	Анталья
12	4	2	Россия	Анталья
13	1	1	Латвия	Москва
14	2	4	Латвия	Москва
15	3	2	Латвия	Москва

16	4	2	Латвия	Москва
17	1	1	Латвия	Сочи
18	2	4	Латвия	Сочи
19	3	2	Латвия	Сочи
20	4	2	Латвия	Сочи
21	1	1	Латвия	Анталья
22	2	4	Латвия	Анталья
23	3	2	Латвия	Анталья
24	4	2	Латвия	Анталья
25	1	1	Турция	Москва
26	2	4	Турция	Москва
27	3	2	Турция	Москва
28	4	2	Турция	Москва
29	1	1	Турция	Сочи
30	2	4	Турция	Сочи
31	3	2	Турция	Сочи
32	4	2	Турция	Сочи
33	1	1	Турция	Анталья
34	2	4	Турция	Анталья
35	3	2	Турция	Анталья
36	4	2	Турция	Анталья

## Cross apply

```

SELECT e.ID,e.CityID, p.Name Country, d.Name City
FROM Hotel AS e
CROSS APPLY City AS d
CROSS APPLY Country AS p

```

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Россия	Москва
3	3	2	Россия	Москва
4	4	2	Россия	Москва
5	1	1	Россия	Сочи
6	2	4	Россия	Сочи
7	3	2	Россия	Сочи
8	4	2	Россия	Сочи
9	1	1	Россия	Анталья
10	2	4	Россия	Анталья
11	3	2	Россия	Анталья
12	4	2	Россия	Анталья
13	1	1	Латвия	Москва
14	2	4	Латвия	Москва
15	3	2	Латвия	Москва
16	4	2	Латвия	Москва
17	1	1	Латвия	Сочи
18	2	4	Латвия	Сочи
19	3	2	Латвия	Сочи
20	4	2	Латвия	Сочи
21	1	1	Латвия	Анталья
22	2	4	Латвия	Анталья
23	3	2	Латвия	Анталья
24	4	2	Латвия	Анталья
25	1	1	Турция	Москва
26	2	4	Турция	Москва
27	3	2	Турция	Москва
28	4	2	Турция	Москва
29	1	1	Турция	Сочи
30	2	4	Турция	Сочи
31	3	2	Турция	Сочи
32	4	2	Турция	Сочи
33	1	1	Турция	Анталья
34	2	4	Турция	Анталья
35	3	2	Турция	Анталья
36	4	2	Турция	Анталья

## Самосоединение

SELECT e.ID,e.CityID, p.Name Country, d.Name City

FROM Hotel AS e, City AS d, Country AS p  
 WHERE d.ID=e.CityID AND p.ID=d.CountryID

	ID	CityID	Country	City
1	1	1	Россия	Москва
2	2	4	Турция	Анталья
3	3	2	Россия	Сочи
4	4	2	Россия	Сочи

## 2. Фильтрация данных в запросах с использованием предикатов (EXISTS, IN, ALL, SOME/ANY, BETWEEN, LIKE).

### EXISTS

SELECT \* FROM City  
 WHERE EXISTS (SELECT \* FROM City WHERE CountryID = 2)

ID	Name	CountryID
----	------	-----------

SELECT \* FROM City  
 WHERE EXISTS (SELECT \* FROM City WHERE CountryID = 1)

	ID	Name	CountryID
1	1	Москва	1
2	2	Сочи	1
3	4	Анталья	3

### IN

SELECT \* FROM City  
 WHERE CountryID IN (2, 3)

	ID	Name	CountryID
1	4	Анталья	3

### ANY

SELECT \* FROM RoomType  
 WHERE Price >= ANY --цена > цены хотя бы одного номера  
 (SELECT Price  
 FROM RoomType  
 WHERE CountOfSeats = 2) -- из тех, у которых кол-во мест = 2

	ID	Price	TypeName	CountOfSe...	CountOfBe...
1	1	2500	Стандарт 2/1	2	1
2	2	2500	Стандарт 2/2	2	2
3	3	2000	Стандарт 1	1	1
4	4	3000	Комфорт	2	1
5	6	1500	Бюджет 2	2	2
6	7	4000	Люкс	2	1

## ALL

```
SELECT * FROM RoomType
WHERE Price >= ALL --цена > всех цен номеров
(SELECT Price
FROM RoomType
WHERE CountOfSeats = 2) -- из тех, у которых кол-во мест = 2
```

	ID	Price	TypeNa...	CountOfSe...	CountOfBe...
1	7	4000	Люкс	2	1

## SOME

```
SELECT * FROM RoomType
WHERE Price >= SOME --цена > цены хотя бы одного номера
(SELECT Price
FROM RoomType
WHERE CountOfSeats = 2) -- из тех, у которых кол-во мест = 2
```

	ID	Price	TypeName	CountOfSe...	CountOfBe...
1	1	2500	Стандарт 2/1	2	1
2	2	2500	Стандарт 2/2	2	2
3	3	2000	Стандарт 1	1	1
4	4	3000	Комфорт	2	1
5	6	1500	Бюджет 2	2	2
6	7	4000	Люкс	2	1

## BETWEEN

```
SELECT *
FROM RoomType
WHERE Price BETWEEN 0 AND 2000 -- типы номеров, цена которых в диапазоне
ORDER BY Price
```

	ID	Price	TypeName	CountOfSe...	CountOfBe...
1	5	1000	Бюджет 1	1	1
2	6	1500	Бюджет 2	2	2
3	3	2000	Стандарт 1	1	1

## LIKE

```
SELECT *  
FROM Client  
WHERE Name LIKE 'И%' -- клиенты, у которых ФИО начинается на 'И'
```

	ID	Name	Passport
1	4	Иванова Екатерина Владиславовна	6348 588685
2	5	Иванов Петр Игоревич	6352 553395

## 3. Запросы с использованием выражения CASE.

```
SELECT  
ID, TypeName, Price,  
CASE  
  WHEN Price >= 3000 THEN 'цена >= 3000'  
  WHEN Price >= 2000 THEN '2000 <= цена < 3000'  
  ELSE 'цена < 2000'  
END PriceRoomTypeWithELSE,
```

```
CASE  
  WHEN Price >= 3000 THEN 'цена >= 3000'  
  WHEN Price >= 2000 THEN '2000 <= цена < 3000'  
END PriceRoomTypeWithoutELSE
```

FROM RoomType

	ID	TypeName	Price	PriceRoomTypeWithEL...	PriceRoomTypeWithoutEL...
1	1	Стандарт 2/1	2500	2000 <= цена < 3000	2000 <= цена < 3000
2	2	Стандарт 2/2	2500	2000 <= цена < 3000	2000 <= цена < 3000
3	3	Стандарт 1	2000	2000 <= цена < 3000	2000 <= цена < 3000
4	4	Комфорт	3000	цена >= 3000	цена >= 3000
5	5	Бюджет 1	1000	цена < 2000	NULL
6	6	Бюджет 2	1500	цена < 2000	NULL
7	7	Люкс	4000	цена >= 3000	цена >= 3000

#### 4. Использование встроенных функций (функций преобразования типов (CAST, CONVERT), функций для проверки значений NULL (ISNULL, NULLIF), COALESCE логических функций (CHOOSE, IIF) ).

##### ISNULL

```
SELECT
*,
CASE ISNULL(BookingID,-1) -- используем замену в случае NULL на -1
WHEN -1 THEN 'Брони не было' -- ID бронирования равным (-1) нет
ELSE 'Бронь была реализована'
END BookingInfo
```

FROM AccommodationInfo

	ID	ClientID	RoomID	Booking...	ArrivalDate	DepartureDate	Cost	BookingInfo
1	1	2	2	2	2022-02-25 12:38:56.077	2022-02-28 11:13:12.997	2000	Бронь была реализована
2	2	3	4	NULL	2022-02-26 19:38:56.077	2022-03-01 11:53:12.997	1000	Брони не было
3	3	4	1	NULL	2022-03-08 16:36:53.990	2022-03-14 10:23:17.987	2500	Брони не было
4	4	5	1	NULL	2022-03-08 16:36:53.990	2022-03-14 10:23:17.987	2500	Брони не было
5	5	2	2	NULL	2022-03-11 18:19:26.877	2022-03-15 09:41:18.700	2000	Брони не было

##### COALESCE логических функций (CHOOSE, IIF) ), CAST

```
SELECT COALESCE (IIF(BookingID IS NULL,'Брони не было', NULL), CHOOSE(1,
CONCAT('ID брони = ', CAST(BookingID AS varchar(2)))))
```

AS Result

FROM AccommodationInfo

	Result
1	ID брони = 2
2	Брони не было
3	Брони не было
4	Брони не было
5	Брони не было



## 5. Запросы с использованием функций для работы со строками (REPLACE, SUBSTRING, STUFF, STR, UNICODE, LOWER, UPPER и т.п.).

### REPLACE

```
SELECT Change = REPLACE(Name, 'По', 'А')
FROM Client
--заменяли Полину на Алину
```

	Change
1	Киреева Маргарита Николаевна
2	Елина Алина Владимировна
3	Иванова Екатерина Владиславовна
4	Иванов Петр Игоревич

### SUBSTRING, UPPER, CHARINDEX

```
SELECT ID,
FirstName=UPPER(SUBSTRING(Name, 1, CHARINDEX(' ', Name)))-- таблица клиентов
только с фамилией заглавными буквами
FROM Client
```

	ID	FirstName
1	2	КИРЕЕВА
2	3	ЕЛИНА
3	4	ИВАНОВА
4	5	ИВАНОВ

### CAST, CONVERT, SUBSTRING

```
SELECT ID, Name,
SeriesPassport=CAST(SUBSTRING(Passport, 1, 5) AS INT),-- таблица с разделением
данных паспорта
NumberPassport=CONVERT(INT, SUBSTRING(Passport, 5, 7))-- на серию и номер
FROM Client
```

	ID	Name	SeriesPassp...	NumberPassp...
1	2	Киреева Маргарита Николаевна	6315	612826
2	3	Елина Полина Владимировна	6311	683459
3	4	Иванова Екатерина Владиславовна	6348	588685
4	5	Иванов Петр Игоревич	6352	553395

## UNICODE

```
SELECT Code = UNICODE(Name)
FROM Client
```

	Code
1	1050
2	1045
3	1048
4	1048

6. Запросы с использованием функций даты и времени (DATEPART, DATEADD, DATEDIFF, GETDATE(), SYSDATETIMEOFFSET() и т.п.).

## DATEPART

```
SELECT ID,
ArrivalMonth=DATEPART(m, ArrivalDate), -- выводим месяц приезда и отъезда
DepartureMonth=DATEPART(m, DepartureDate)
FROM AccommodationInfo
```

	ID	ArrivalMonth	DepartureMonth
1	1	2	2
2	2	2	3
3	3	3	3
4	4	3	3
5	5	3	3

## DATEDIFF

```
SELECT ID,
LengthOfAccommodation=DATEDIFF(d, ArrivalDate, DepartureDate) -- продолжительность
проживания в днях
FROM AccommodationInfo
```

	ID	LengthOfAccommodat...
1	1	3
2	2	3
3	3	6
4	4	6
5	5	4

## GETDATE()

	CurrentDate
1	2022-03-21 21:01:46.380

## SYSDATETIMEOFFSET()

	CurrentDate
1	2022-03-21 21:02:48.1172083 +04:00

## 7. Запросы с использованием агрегатных функций, группировок (GROUP BY) и фильтрации групп (HAVING).

### Запросы с использованием агрегатных функций

```
SELECT  
COUNT(*) [Общее кол-во типов номеров],  
MAX(Price) [Максимальная цена номера],  
MIN(Price) [Минимальная цена номера],  
AVG(Price) [Средняя цена номера]  
FROM RoomType
```

	Общее кол-во типов номеров	Максимальная цена номера	Минимальная цена номера	Средняя цена номера
1	7	4000	1000	2357

## GROUP BY, HAVING

```
SELECT  
ClientID, d.Name,  
COUNT(*) VisitsCount, --кол-во посещений отеля  
SUM(Cost) SumCost, --сумма, потраченная на проживание, для каждого клиента за  
всё время  
AVG(Cost) AverageCost -- средняя стоимость проживания для каждого клиента  
FROM AccommodationInfo  
LEFT JOIN Client AS d ON d.ID=ClientID  
GROUP BY ClientID, d.Name  
HAVING SUM(Cost)>2000 --сумма больше 2000  
ORDER BY ClientID
```

	ClientID	Name	VisitsCo...	SumCost	AverageC...
1	2	Киреева Маргарита Николаевна	2	4000	2000
2	4	Иванова Екатерина Владиславовна	1	2500	2500
3	5	Иванов Петр Игоревич	1	2500	2500