



# SQL Basic

Lecturer: Nataliya Bogushevskaya, Kyiv

E-mail:  
[Nataliya\\_Bogushevskaya@epam.com](mailto:Nataliya_Bogushevskaya@epam.com)

---

# Agenda

- Несколько слов о SQL
- Простые запросы SELECT ... FROM
- Сортировка данных
- Секция WHERE
  - Логические операции
  - Like
  - Between ... And
  - IN
  - Case

# ОСНОВЫ SQL

# Structured query language

SQL состоит из набора команд манипулирования данными в реляционной БД, которые позволяют

- создавать объекты реляционной базы данных,
- модифицировать данные в таблицах (вставлять, удалять, исправлять),
- изменять схемы отношений БД,
- выполнять вычисления над данными,
- делать выборки из БД,
- поддерживать безопасность и целостность данных.

# Основные операторы SQL

Весь набор команд SQL можно разбить на следующие группы:

- команды определения данных (DDL – Data Definition Language);
- команды манипулирования данными (DML – Data Manipulation Language);
- команды выборки данных (DQL– Data Query Language);
- команды управления транзакциями;
- команды управления данными.

# DDL

|                |  |
|----------------|--|
| ALTER TABLE    | Изменяет описание таблицы                |
| CREATE INDEX   | Создает индекс для таблицы               |
| CREATE TABLE   | Определяет таблицу                       |
| CREATE TRIGGER | Создает триггер в БД                     |
| CREATE VIEW    | Определяет представление на таблицах     |
| DROP INDEX     | Физически удаляет индекс из баз данных   |
| DROP TABLE     | Физически удаляет таблицу из базы данных |
| DROP VIEW      | Удаляет представление                    |

# DML

|        |  |
|--------|--|
| DELETE | Удаляет одну или более строк из таблицы базы данных  |
| INSERT | Вставляет одну или более строк в таблицу базы данных |
| UPDATE | Обновляет значения колонок в таблице базы данных     |

# DQL

|           |   |
|-----------|---|
| SELECT    | Выполняет запрос на выборку данных из таблиц и представлений                      |
| UNION     | Объединяет в одной выборке результаты выполнения двух или более команд SELECT     |
| MINUS     | Результаты выполнения команды SELECT кроме результатов выполнения второй команды. |
| INTERSECT | Пересечение в одной выборке результатов выполнения двух или более команд SELECT   |



# Команды управления транзакциями

|           |  |
|-----------|--|
| COMMIT    | Завершает транзакцию и физически актуализирует состояние базы данных   |
| ROLLBACK  | Завершает транзакцию и возвращает текущее состояние базы данных на момент последней завершенной транзакции и контрольной точки |
| SAVEPOINT | Назначает контрольную точку внутри транзакции  |

# Простые запросы

# Простые запросы

Оператор `SELECT` – один из наиболее важных и используемых операторов SQL. Он позволяет производить выборки данных из БД и преобразовывать к нужному виду полученные результаты.

# Простейший вид оператора


В простейшем случае оператор SELECT имеет вид:

```
SELECT { * | <поле1> [, <поле2> ...] }  
FROM <таблица1>
```

# Простейший вид оператора

```
select *  
from HumanResources.Employee;
```

```
select NationalIDNumber  
        , LoginID  
        , JobTitle  
        , Gender  
        , HireDate  
from HumanResources.Employee
```

| Employee (HumanResources)   |                   |
|---|-------------------|
|  | BusinessEntityID  |
|   | NationalIDNumber  |
|   | LoginID           |
|   | OrganizationNode  |
|   | OrganizationLevel |
|   | JobTitle          |
|   | BirthDate         |
|   | MaritalStatus     |
|   | Gender            |
|   | HireDate          |
|   | SalariedFlag      |
|   | VacationHours     |
|   | SickLeaveHours    |
|   | CurrentFlag       |
|   | rowguid           |
|   | ModifiedDate      |

# Простейший вид оператора

Для устранения повторяющихся записей используется оператор DISTINCT:

```
select distinct Gender  
from HumanResources.Employee
```

# Простейший вид оператора

В списке могут использоваться не только поля, но и любые выражения от них с арифметическими операциями +, -, \*, /. После выражения может записываться псевдоним выражения в форме **AS** **<псевдоним>**.

# Простейший вид оператора

```
select HireDate, getdate() as Now  
from HumanResources.Employee
```

```
select HireDate, getdate() as Now  
        ,DATEDIFF(year, HireDate, getdate()) as  
experience  
from HumanResources.Employee
```



# Псевдонимы таблиц

В запросе SELECT можно объединить данные нескольких таблиц.

Каждое имя поля должно предваряться **ссылкой** на таблицу, к которой она относится. В операторах, работающих с несколькими таблицами, обычно каждой таблице даётся **псевдоним**, сокращающий ссылки на таблицы.

```
select p.FirstName + ' ' + p.LastName as Surname  
from [Person].[Person] p
```

Алиас (псевдоним)

# Полезные функции

|   |   |
|---|---|
| DATEDIFF ( datepart ,<br>startdate ,<br>enddate ) | Разница дат                               |
| DATENAME ( datepart ,<br>date )                   | Возвращает строку с указанной частью даты |
| GETDATE ( )                                       | Системная (текущая) дата                  |
| YEAR(date )                                       | Возвращает год из даты                    |

| datepart  | Сокращения | datepart    | Сокращения |
|-----------|------------|-------------|------------|
| year      | yy, yyyy   | hour        | hh         |
| quarter   | qq, q      | minute      | mi, n      |
| month     | mm, m      | second      | ss, s      |
| dayofyear | dy, y      | millisecond | ms         |
| day       | dd, d      | microsecond | mcs        |
| week      | wk, ww     | nanosecond  | ns         |
| weekday   | dw, w      |             |            |

# Задания

1. Выбрать список должностей (HumanResources.Employee поле JobTitle)
2. Выбрать 3П сотрудников как произведение Rate и PayFrequency (таблица EmployeePayHistory)
3. Вывести день недели своего дня рождения.
4. Найти, сколько дней осталось до конца года

# Сортировка данных

Набор данных, выдаваемый в результате выполнения оператора `SELECT`, в общем случае возвращается в неотсортированном виде.

Определить, по каким полям необходимо сортировать записи в результирующем наборе данных, можно при помощи оператора `ORDER BY`, имеющего следующий формат:

**`ORDER BY`** <столбец>[`asc` | `desc`]


# Сортировка данных

```
select FirstName,MiddleName,LastName  
from [Person].[Person]  
order by LastName, MiddleName desc
```

|   | First Name | Middle Name | Last Name   |
|---|------------|-------------|-------------|
| 1 | Syed       | E           | Abbas       |
| 2 | Catherine  | R.          | Abel        |
| 3 | Kim        | B           | Abercrombie |
| 4 | Kim        | NULL        | Abercrombie |
| 5 | Kim        | NULL        | Abercrombie |

# Задание

- Вывести список занимаемых должностей в алфавитном порядке

| Employee (HumanResources)   |                   |
|---|-------------------|
|  | BusinessEntityID  |
|   | NationalIDNumber  |
|   | LoginID           |
|   | OrganizationNode  |
|   | OrganizationLevel |
|   | JobTitle          |
|   | BirthDate         |
|   | MaritalStatus     |
|   | Gender            |
|   | HireDate          |
|   | SalariedFlag      |
|   | VacationHours     |
|   | SickLeaveHours    |
|   | CurrentFlag       |
|   | rowguid           |
|   | ModifiedDate      |

# Использование WHERE

С использованием оператора WHERE оператор SELECT имеет следующий формат:

**SELECT** { \* | <поле1> [, <поле2> ...] }

**FROM** <таблица1>

**WHERE** <условия поиска>

# Использование WHERE

В набор данных, который возвращается как результат выполнения оператора SELECT, будут включаться только те записи, которые удовлетворяют условию WHERE. Условие может включать:

- имена полей (кроме вычисляемых),
- константы,
- логические выражения с арифметическими операциями,
- логические операции **and**, **or**, **not**,



# Операции отношения

```
select *  
from [Person].[Person] p  
where p.FirstName = 'Kim'
```

```
select JobTitle, Gender  
from [HumanResources].[Employee]  
where year([HireDate]) = '2001'
```

|        |                  |
|--------|------------------|
| =      | равно            |
| >      | больше           |
| >=, !< | больше или равно |
| <      | меньше           |
| <=, !> | меньше или равно |
| !=, <> | не равно         |

# Логические операции

| AND   | TRUE  | FALSE | NULL |
|-------|-------|-------|------|
| TRUE  | true  | false |      |
| FALSE | false | false |      |
| NULL  |       |       |      |

| OR    | TRUE | FALSE | NULL |
|-------|------|-------|------|
| TRUE  | true | true  |      |
| FALSE | true | false |      |
| NULL  |      |       |      |

**/\*Выбрать всю информацию о людях, у которых есть второе имя, упорядочить список по фамилии\*/**

```
select *  
from [Person].[Person] p  
where p.MiddleName is not null  
order by p.LastName
```

# Задания

1. Вывести Логин сотрудников, у которых часы отпуска  $> 20$ , и которые работают на должности 'Buyer'.
2. Вывести Логин сотрудников, у которых SickLeaveHours больше, чем 20% от VacationHours
3. Вывести Логин и возраст неженатых мужчин, упорядочив по возрасту (от самых старых к самым юным).

# Like

Операция Like имеет следующий синтаксис:

***Field [NOT] LIKE pattern [ESCAPE esc\_char ]***

| Wildcard character | Description   |
|--------------------|---|
| %                  | Любая подстрока   |
| _                  | Один любой символ   |
| [ ]                | Любой символ из множества ([a-f]) - это множество ([abcdef]). |
| [^]                | Любой символ не из указанного множества                       |

# Like

**/\*Выбрать телефоны, начинающиеся на 415\*/**

select \*

from Person.PersonPhone

where PhoneNumber **like** '415%'

**/\*Выбрать телефоны, не начинающиеся на 415\*/**

select \*

from Person.PersonPhone

where PhoneNumber **not like** '415%'

# Like

```
SELECT distinct FirstName  
FROM Person.Person  
where FirstName like 'Andre[aw]'  
order by FirstName
```

| FirstName |
|-----------|
| Andrea    |
| Andrew    |

```
SELECT distinct FirstName  
FROM Person.Person  
where FirstName like 'Andre_'  
order by FirstName
```

| FirstName |
|-----------|
| Andrea    |
| Andres    |
| Andrew    |

```
SELECT distinct FirstName  
FROM Person.Person  
where FirstName like 'Andre%'  
order by FirstName
```

| FirstName |
|-----------|
| Andre     |
| Andrea    |
| Andreas   |
| Andres    |
| Andrew    |

ХМ...

-- Выбрать сотрудников, email которых  
содержит символ «\_»

--неверно!

```
select *  
from HumanResources.vEmployee  
where EmailAddress like '%_%'
```

**--верно**

```
select *  
from HumanResources.vEmployee  
where EmailAddress like '%A_%' escape 'A'
```

# between ... and

Операция `between ... and` имеет синтаксис:

**`<поле> between <значение> and <значение>`**

и задаёт для указанного поля диапазон отбираемых значений.

`select *`

`from HumanResources.Employee`

`where VacationHours between 10 and 20`





***in***

Операция ***in*** имеет синтаксис:

**<поле> *in* (<множество>)**

и отбирает записи, в которых значение указанного поля является одним из элементов указанного множества.

in

```
select e.LoginID
from HumanResources.Employee e
WHERE e.JobTitle = 'Design Engineer'
      OR e.JobTitle = 'Tool Designer'
      OR e.JobTitle = 'Marketing Assistant';
```

```
select e.LoginID
from HumanResources.Employee e
WHERE e.JobTitle in ('Design Engineer',
                    'Tool Designer',
                    'Marketing Assistant');
```

# Case

СИНТАКСИС:

```
CASE input_expression  
    WHEN expression1 THEN result1  
    WHEN expression1 THEN result1  
    ...  
    ELSE else_result  
END
```

```
select Gender,  
       case  
         when Gender = 'M' then 'Male'  
         when Gender = 'F' then 'Female'  
         else 'Unknown'  
       end as Gender  
from [HumanResources].[Employee]
```

|   | Gender | Gender |
|---|--------|--------|
| 1 | M      | Male   |
| 2 | F      | Female |
| 3 | M      | Male   |

# Задания

1. Вывести Логин неженатых мужчин в возрасте от 25 до 45 и должностью «Менеджер» (любой менеджер)
2. Найти все страны, состоящие более, чем одного слова (разделителем может быть пробел или «-») – таблица Person.CountryRegion.
3. Найти людей, у которых или нет второго имени, или оно состоит из одной буквы (учитывать “J” и “J.”) – таблица Person.Person.
4. Вывести номер продукта и его категорию (таблица Production.Product, поля ProductNumber и ProductLine), где категорию отображать в виде: 'R' - 'Road', 'M' - 'Mountain', 'T' - 'Touring', 'S' - 'Other sale items' и 'Not for sale' в остальных случаях.