

Давайте знакомиться 😊

Айсет Алдиев

1. Практикующий преподаватель.
2. Опыт в телекоме около 8 лет.
3. Опыт в банковской сфере более 2 лет
4. [LinkedIn](#)



Что разберем сегодня

1. Введение в базы данных
2. Типы БД
3. Инструментарий курса
4. Список возможностей инструкций SQL SELECT
5. Инструкции SELECT, FROM, WHERE, ORDER BY

Введение в Базы данных

Зачем нужны БД

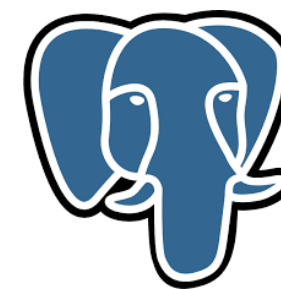
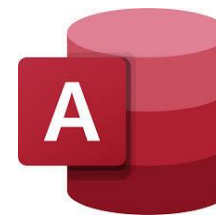
- Основная функция базы данных – предоставление единого хранилища для всей информации, относящейся к определенной теме.
- Вместо того чтобы выискивать нужные сведения в документах Word, таблицах Excel, текстовых файлах, сообщениях электронной почты и самоклеющихся заметках, их можно взять из единой базы.
- База данных может содержать все что угодно, будь-то список приглашенных на свадьбу гостей или информация о каждом клиенте, посетившем Web-сайт электронного магазина и разместившего там свои заказы.

Основные понятия

- Базы данных (*БД*) - совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
- Пример: БД о наличии медикаментов, БД документов учеников школы, картотека отдела кадров, БД правительства РК и т.д.

Типы БД

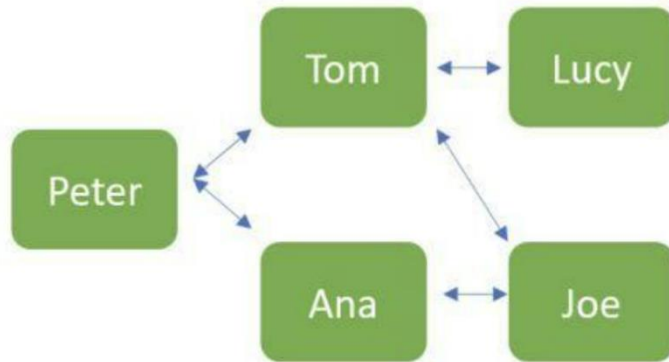
- Файл-серверные: Microsoft Access
- Клиент-серверные: MySQL, PostgreSQL, Oracle DB
- Встраиваемые: SQLite



Что такое СУБД?

- СУБД - это программа, с помощью которой осуществляется хранение, обработка и поиск информации в базе данных.
- СУБД бывают как реляционные так и не реляционные.
- Реляционные: Oracle DB, PostgreSQL, MS Sql.
- Не реляционные, ещё по другому NoSQL БД, можно отнести
MongoDB, Cassandra, HBase

Отличие NoSQL и SQL



NODE	FRIENDS
Peter	{Tom, Ana}
Tom	{Joe, Lucy, Peter}
Ana	{Joe, Lucy, Peter}
Lucy	{Tom}
Joe	{Tom, Ana}

```
var cars = [  
  { Model: "BMW", Color: "Red", Manufactured: 2016 },  
  { Model: "Mercedes", Type: "Coupe", Color: "Black", Manufactured: "1-1-2017" }  
];
```

```
SELECT Orders.OrderID, Customers.Name, Orders.Date  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustID = Customers.CustID
```

Функции СУБД

- ввод
- хранение
- манипулирование
- обработку запросов к БД
- поиск
- выборку
- сортировку
- обновление
- защиту данных от несанкционированного доступа или потери

Реляционная СУБД

- Почему именно реляционную СУБД?
- Реляционная алгебра, на базе этой теории строятся все современные СУБД. Реляционная алгебра определяет систему операции над отношениями (таблицами): объединение, деление, вычитание, соединение.
- Это можно сказать на языке математики:
+ (плюс) - (минус) * (умножение) / (деление)
- Все эти операции выражаются на языке запросов SQL

Реляционная модель

Модель представляет собой фиксированную структуру математических понятий, которая описывает то, как будут представлены данные. Базовой единицей данных в пределах реляционной модели является таблица.

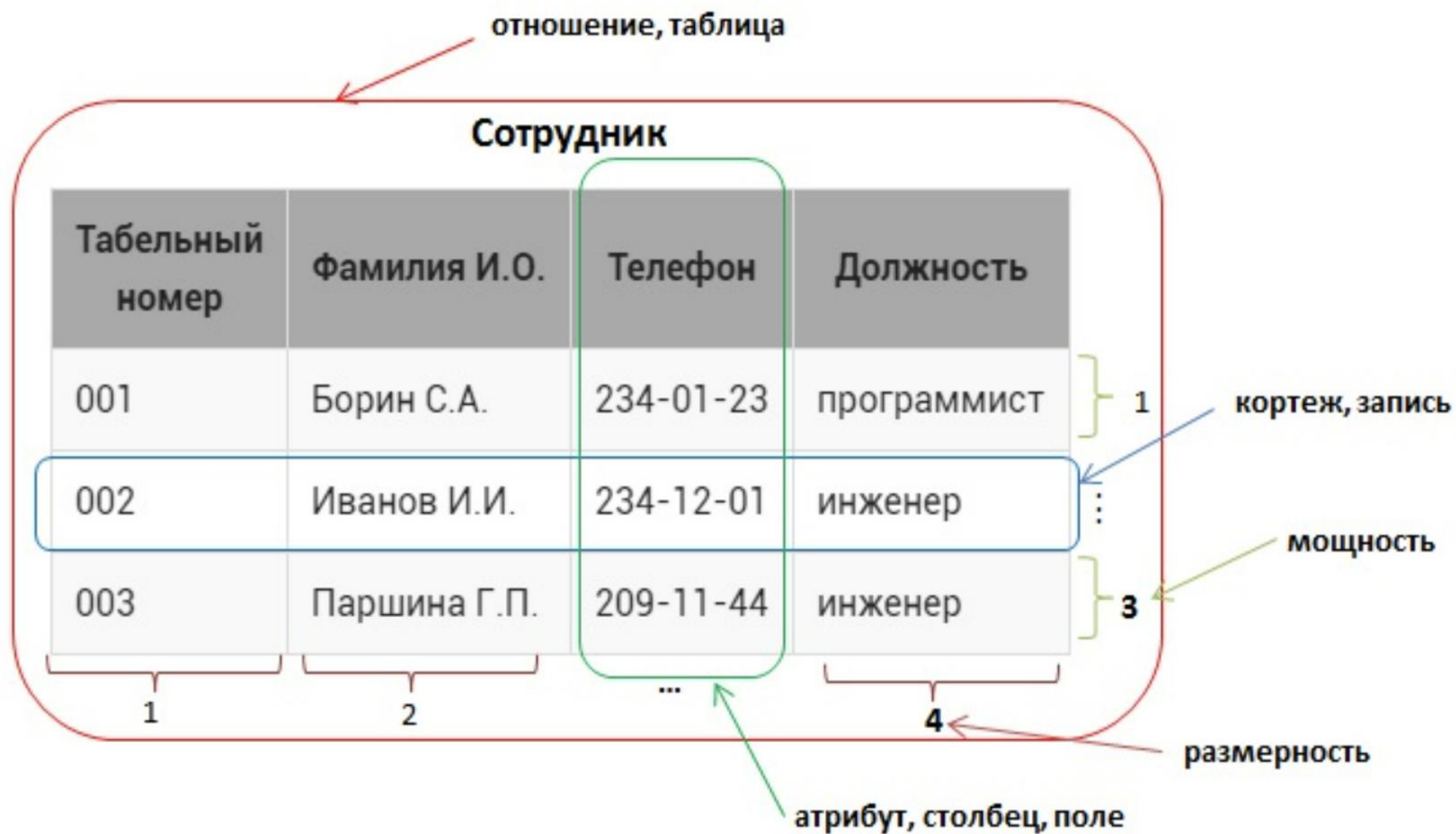
- Сущность (entity)
- Столбец (column)
- Строка (row)
- Таблица (table)
- Первичный ключ (primary key)
- Внешний ключ (foreign key)

Клиент		
cust_id	fname	lname
1	Джордж	Блейк
2	Сью	Смит

Счет			
account_id	product_cd	cust_id	balance
103	CHK	1	\$75.00
104	SAV	1	\$250.00
105	CHK	2	\$783.64
106	MM	2	\$500.00
107	LOC	2	0

Тип счета	
product_cd	name
CHK	Текущие расходы
SAV	Сбережения
MM	Денежный рынок
LOC	Кредитный лимит

Транзакция				
txn_id	txn_type_cd	account_id	amount	date
978	DBT	103	\$100.00	2004-01-22
979	CDT	103	\$25.00	2004-02-05
980	DBT	104	\$250.00	2004-03-09
981	DBT	105	\$1000.00	2004-03-25
982	CDT	105	\$138.50	2004-04-02
983	CDT	105	\$77.86	2004-04-04
984	DBT	106	\$500.00	2004-03-27



Какие ещё есть модели БД

Иерархическая **модель** данных;

Сетевая **модель**;

ER-**модель**;

Расширенная **модель** сущностных отношений;

Объектная **модель**;

Документоориентированная **модель**;

Модель объекта-атрибута;

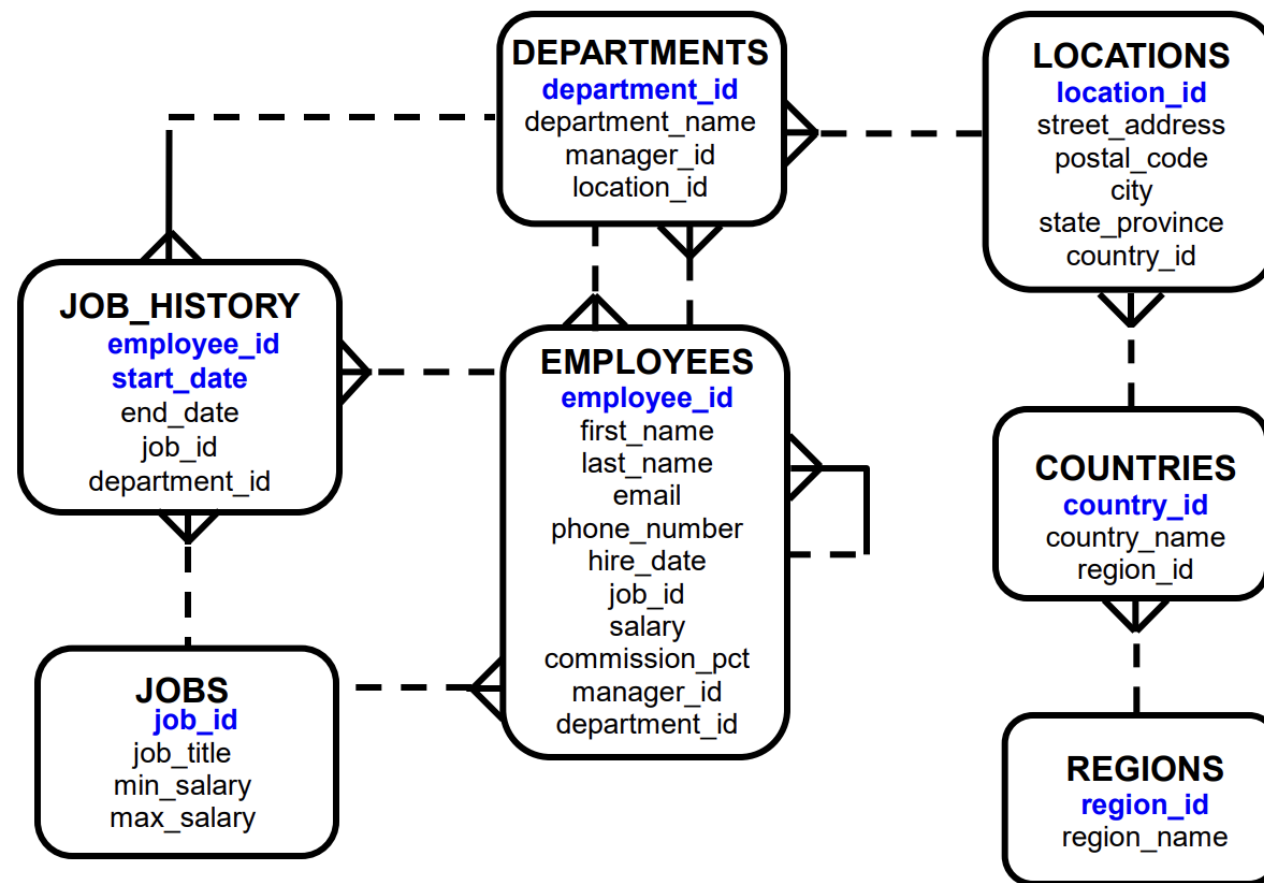
Инструментарий курса

С чем мы будем работать?

1. Oracle DB – СУБД.
2. Зарегистрируемся на сайте [livesql.oracle](https://livesql.oracle.com) для работы с БД.
3. Напишем “Hello World!”
4. Познакомимся с БД HR.
5. IDE: SQL Developer, PL SQL Developer, TOAD

БД HR

The Human Resources (HR) Schema



Поговорим про SQL

Введение в SQL

- SQL это непроцедурный язык
- Процедурные языки нужны для реализации какой-либо логики
- SQL это декларативный язык (туда же входит HTML)

```
1 | SELECT * FROM Users WHERE Country='Mexico';
```

```
1 | <article>
2 |   <header>
3 |     <h1>Declarative Programming</h1>
4 |     <p>Sprinkle Declarative in your verbiage to sound smart</p>
5 |   </header>
6 | </article>
```

Инструкции SQL

SELECT
INSERT
UPDATE
DELETE
MERGE

Язык манипулирования данными (DML)

CREATE
ALTER
DROP
RENAME
TRUNCATE
COMMENT

Язык определения данных (DDL)

GRANT
REVOKE

Язык управления данными (DCL)

COMMIT
ROLLBACK
SAVEPOINT

Управление транзакциями (TCL)

- Результатом SQL запроса является результирующий набор.
- Язык SQL делится на команды DDL, DML, TCL, DCL
- DDL (Data Definition Language) – CREATE, ALTER, DROP, TRUNCATE
- DML (Data Manipulation Language) – SELECT, INSERT, UPDATE, DELETE
- TCL (Transaction Control Language) – COMMIT, ROLLBACK, SAVEPOINT
- DCL (Data Control Language) – GRANT, REVOKE
- ANSI (1989, 1992, 1999 и 2003 гг.)
- Процедурный SQL. PL/SQL, PostgreSQL, TransactSQL.

Блок	Назначение
Select	Определяет столбцы, которые должны быть включены в результирующий набор запроса
From	Указывает таблицы, из которых должны быть извлечены данные, и то, как эти таблицы должны быть соединены
Where	Ограничивает число строк в окончательном результирующем наборе
Group by	Используется для группировки строк по одинаковым значениям столбцов
Having	Ограничивает число строк в окончательном результирующем наборе с помощью группировки данных
Order by	Сортирует строки окончательного результирующего набора по одному или более столбцам

Типы данных Oracle DB

- CHAR, VARCHAR2
- NUMBER
- DATE
- LOB, CLOB, BLOB, XMLTYPE

Выбор всех столбцов

SQL Worksheet

```
1 select * from account;
```

ACCOUNT_ID	AVAIL_BALANCE	CLOSE_DATE	LAST_ACTIVITY_DATE	OPEN_DATE	PENDING_BALANCE	STATUS	CUST_ID	OPEN_BRANCH_ID	OPEN_EMP_ID	PRODUCT_CD
1	1057.75	-	04-JAN-05	15-JAN-00	1057.75	ACTIVE	1	2	10	CHK
2	500	-	19-DEC-04	15-JAN-00	500	ACTIVE	1	2	10	SAV
3	3000	-	30-JUN-04	30-JUN-04	3000	ACTIVE	1	2	10	CD
4	2258.02	-	27-DEC-04	12-MAR-01	2258.02	ACTIVE	2	2	10	CHK

Выбор конкретных столбцов

```
select account_id, status from account;
```

ACCOUNT_ID	STATUS
1	ACTIVE
2	ACTIVE
3	ACTIVE
4	ACTIVE
5	ACTIVE
6	ACTIVE
7	ACTIVE

Арифметические выражения

Получение выражений с данными в числовом формате или формате дат с помощью арифметических операторов

Оператор	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление

Использование арифметических операторов

```
SELECT  
account_id, avail_balance, avail_balance + 1000 as balance_2  
FROM account
```

ACCOUNT_ID	AVAIL_BALANCE	BALANCE_2
1	1057.75	2057.75
2	500	1500

Приоритет операторов

```
SELECT  
account_id, avail_balance, 2 * avail_balance + 1000  
FROM account
```

ACCOUNT_ID	AVAIL_BALANCE	2*AVAIL_BALANCE+1000
1	1057.75	3115.5

```
SELECT  
account_id, avail_balance, 2 * (avail_balance + 1000)  
FROM account
```

ACCOUNT_ID	AVAIL_BALANCE	2*(AVAIL_BALANCE+1000)
1	1057.75	4115.5

Использование псевдонимов столбцов

```
SELECT  
account_id, avail_balance, avail_balance + 1000 as balance_2  
FROM account
```

ACCOUNT_ID	AVAIL_BALANCE	BALANCE_2
1	1057.75	2057.75
2	500	1500

Оператор конкатенации

- «Склеивает» строки
- Обозначается двумя вертикальными палками ||

```
SELECT  
first_name || last_name as emp_name  
FROM employee
```

EMP_NAME
MichaelSmith
SusanBarker

Использование символьных строк литералов

```
SELECT  
first_name || ' is a ' || title as emp_position  
FROM employee
```

EMP_POSITION
Michael is a President
Susan is a Vice President
Robert is a Treasurer

Дубли строк

```
select distinct product_cd from account;
```

PRODUCT_CD
MM
SAV
BUS
SBL
CHK
CD

- Средства ограничения числа строк:
 - Предложение WHERE
 - Условия сравнения с операторами
=, <=, BETWEEN, IN, LIKE и NULL
 - Логические условия с операторами AND, OR и NOT
- Правила приоритета операторов, используемых в выражении
- Сортировка строк с использованием предложения ORDER BY

Ограничение числа выбранных строк

- Ограничение набора возвращаемых строк с помощью предложения WHERE:

```
SELECT *|{[DISTINCT] столбец|выражение [псевдоним],...}  
FROM таблица [WHERE условия];
```

- Предложение WHERE следует за предложением FROM.

Использование выражения WHERE

```
select emp_id, first_name, last_name, start_date, title  
FROM employee  
where emp_id = 1
```

EMP_ID	FIRST_NAME	LAST_NAME	START_DATE	TITLE
1	Michael	Smith	22-JUN-01	President

Символьные строки и даты

- Символьные строки и значения дат заключаются в одиночные кавычки.
- Символьные значения проверяются с учетом регистра, а значения дат – с учетом формата.

```
select emp_id, first_name, last_name, start_date, title  
FROM employee  
where first_name = 'Michael';
```

```
select emp_id, first_name, last_name, start_date, title  
FROM employee  
where start_date = '22-JUN-01';
```

Операторы сравнения

=	Равно
>	Больше
>=	Больше или равно
<	Меньше
<=	Меньше или равно
<>	Не равно
BETWEEN ...AND...	Между двумя значениями
IN (set)	Совпадает с любым значением из списка
LIKE	Соответствует шаблону символов
IS NULL	Является значением null

Использование операторов сравнения

```
select account_id, avail_balance from account  
where avail_balance < 1000;
```

ACCOUNT_ID	AVAIL_BALANCE
2	500
5	200
8	534.12

Условия диапазона, использующие оператор BETWEEN

Оператор BETWEEN используется для вывода строк на основе диапазона значений:

```
select account_id, avail_balance from account  
where avail_balance between 1000 and 2000;
```

ACCOUNT_ID	AVAIL_BALANCE
1	1057.75
6	1057.75
19	1500

Сопоставление с шаблоном, использующее оператор LIKE

- Оператор LIKE применяется при поиске допустимых строковых значений с использованием подстановочных знаков.
- Условие поиска может включать символьный или числовой литерал:
 - `%` обозначает произвольное число символов (в том числе, возможно, нулевое).
 - `_` обозначает строго один символ.

```
select emp_id, first_name, last_name, start_date, title  
FROM employee  
where first_name like 'M%';
```

```
SELECT last_name  
FROM employee  
WHERE last_name LIKE '_a%e%';
```


Поиск строк, содержащих % или _

Идентификатор ESCAPE позволяет искать сами символы % и _.

```
select  
name  
from  
(select 'Smith_' as name from dual  
)  
where name like '_m%?_' ESCAPE '?'
```

NAME
Smith_

Использование условий NULL

Проверка на наличие пустых значений (null) производится с помощью оператора IS NULL.

```
SELECT  
emp_id, first_name, last_name, start_date, title  
FROM employee  
WHERE end_date IS NULL;
```

EMP_ID	FIRST_NAME	LAST_NAME	START_DATE	TITLE
1	Michael	Smith	22-JUN-01	President
2	Susan	Barker	12-SEP-02	Vice President

Определение условий с использованием логических операторов

AND	Возвращает TRUE, если <i>оба</i> составляющих условия истинны
OR	Возвращает TRUE, если <i>любое</i> из составляющих условий истинно
NOT	Возвращает TRUE, если условие ложно

Использование оператора AND

В случае оператора AND необходимо, чтобы оба составляющих условия были истинными:

```
select first_name, last_name, start_date  
from employee  
where start_date > '17-MAR-04'  
and first_name like 'C%';
```

FIRST_NAME	LAST_NAME	START_DATE
Chris	Tucker	15-SEP-04

Использование оператора OR

В случае оператора OR необходимо, чтобы любое из составляющих условий было истинным:

```
select first_name, last_name, start_date  
from employee  
where start_date > '17-MAR-01'  
or first_name like 'C%';
```

FIRST_NAME	LAST_NAME	START_DATE
Michael	Smith	22-JUN-01
Susan	Barker	12-SEP-02
Susan	Hawthorne	24-APR-02
John	Gooding	14-NOV-03

Использование оператора NOT

```
SELECT first_name, last_name, start_date, title
FROM employee
WHERE title NOT IN ('President', 'Loan Manager', 'Teller');
```

FIRST_NAME	LAST_NAME	START_DATE	TITLE
Susan	Barker	12-SEP-02	Vice President
Robert	Tyler	09-FEB-00	Treasurer
Susan	Hawthorne	24-APR-02	Operations Manager
Helen	Fleming	17-MAR-04	Head Teller

Правила приоритета

```
SELECT
first_name,
last_name,
title,
start_date
FROM employee
WHERE title = 'Teller'
AND first_name LIKE 'S%'
OR start_date > '22-JUN-01';
```

FIRST_NAME	LAST_NAME	TITLE	START_DATE
Susan	Barker	Vice President	12-SEP-02
Susan	Hawthorne	Operations Manager	24-APR-02
John	Gooding	Loan Manager	14-NOV-03

```
SELECT
first_name,
last_name,
title,
start_date
FROM employee
WHERE title = 'Teller'
AND (first_name LIKE 'S%'
OR start_date > '22-JUN-01'
);
```

FIRST_NAME	LAST_NAME	TITLE	START_DATE
Chris	Tucker	Teller	15-SEP-04
Sarah	Parker	Teller	02-DEC-02
Jane	Grossman	Teller	03-MAY-02

Использование предложения ORDER BY

- Сортировка извлеченных строк с помощью предложения ORDER BY:
 - ASC: в порядке возрастания (по умолчанию)
 - DESC: в порядке убывания
- Предложение ORDER BY должно быть последним в инструкции SELECT:

```
select  
account_id,  
avail_balance,  
open_date  
from account  
order by open_date
```

ACCOUNT_ID	AVAIL_BALANCE	OPEN_DATE
1	1057.75	15-JAN-00
2	500	15-JAN-00
9	767.77	15-JAN-00

Сортировка

- Сортировка в порядке убыванию:

```
select  
account_id,  
avail_balance,  
open_date  
from account  
order by open_date DESC;
```

- Сортировка по псевдониму столбца:

```
select  
account_id,  
avail_balance,  
open_date  
from account  
order by avail_balance;
```

Сортировка

- Сортировка по номеру позиции столбца:

```
select
account_id,
avail_balance,
open_date
from account
order by 3 desc;
```

- Сортировка по нескольким столбцам:

```
select
account_id,
avail_balance,
open_date
from account
order by account_id, open_date;
```

ВОПРОСЫ